

Atílio Cláudio Hsu

Gerador Adaptativo de Malhas 2-D para Problemas Eletromagnéticos

Dissertação de Mestrado apresentada à Faculdade de Engenharia Elétrica e de Computação como parte dos requisitos para obtenção do título de Mestre em Engenharia Elétrica. Área de concentração: Telecomunicações e Telemática.

Orientador: Prof. Dr. Hugo Enrique Hernández Figueroa

Campinas, SP
2007

Atílio Cláudio Hsu

Gerador Adaptativo de Malhas 2-D para Problemas Eletromagnéticos

Dissertação de Mestrado apresentada à Faculdade de Engenharia Elétrica e de Computação como parte dos requisitos para obtenção do título de Mestre em Engenharia Elétrica. Área de Concentração: Telecomunicações e Telemática.

Orientador: Prof. Dr. Hugo Enrique Hernández Figueroa

Banca examinadora:

Prof. Dr. Hugo Enrique Hernández Figueroa – DMO/FEEC/UNICAMP

Prof. Dr. Rui Fragassi Souza – DMO/FEEC/UNICAMP

Prof. Dr. Leonardo Lorenzo Bravo Roger – DTT/CESET/UNICAMP

Prof. Dr. Vitaly Félix Rodríguez Esquerre – DTEE/CEFET-BA

Campinas, SP
2007

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA E ARQUITETURA - BAE - UNICAMP

H859g Hsu, Atílio Cláudio
Gerador adaptativo de malhas 2-D para problemas
eletromagnéticos / Atílio Cláudio Hsu. --Campinas, SP:
[s.n.], 2007.

Orientador: Hugo Enrique Hernández Figueroa
Dissertação (Mestrado) - Universidade Estadual de
Campinas, Faculdade de Engenharia Elétrica e de
Computação.

1. Geração numérica de malhas (Análise numérica). 2.
Triangulação. 3. Programação orientada a objetos. 4.
Método dos elementos finitos. I. Hernández Figueroa, Hugo
Enrique. II. Universidade Estadual de Campinas. Faculdade
de Engenharia Elétrica e de Computação. III. Título.

Título em Inglês: Adaptive 2-D mesh generator for electromagnetic problems

Palavras-chave em Inglês: Mesh generation, Delaunay triangulation, Object oriented
programming, Finite element method

Área de concentração: Telecomunicações e Telemática

Titulação: Mestre em Engenharia Elétrica

Banca examinadora: Rui Fragassi Souza, Leonardo Lorenzo Bravo Roger e Vitaly Félix
Rodríguez Esquerre

Data da defesa: 28/05/2007

Programa de Pós-Graduação: Engenharia Elétrica

Resumo

Nesta dissertação apresenta-se o desenvolvimento e a implementação de algoritmos para geração de malhas adaptativas de elementos triangulares, visando o estudo de problemas de propagação eletromagnética em domínios bidimensionais. Foram implementadas duas técnicas de refinamento, uma baseada no método da troca de arestas denominado Delaunay e a outra baseada na técnica de suavização de Laplace. O gerador implementado leva em conta interfaces de curvatura arbitrária e, também, pode ser facilmente integrado aos programas desenvolvidos no Departamento de Microonda e Óptica (DMO).

Palavras-chave: Geração de Malhas, Triangulação de Delaunay, Programação Orientada a Objetos, Método dos Elementos Finitos.

Abstract

This work presents the development of algorithms for adaptive mesh generation of triangular elements applicable to electromagnetic propagation problems in bidimensional domains. Several techniques were used, such as edge-flip, known as Delaunay method, and the Laplacian's smoothing method. The program considers arbitrary curvature interfaces and can be easily integrated to the programs created in the Department of Microwaves and Optics (DMO).

Key words: Mesh Generation, Delaunay Triangulation, Object Oriented Programming, Finite Element Method.

Aos meus queridos pais Chang e Eliza
e à minha querida irmã Lígia.

Agradecimentos

Aqui manifesto meus sinceros agradecimentos ao meu orientador Prof. Dr. Hugo Enrique Hernández Figueroa, pela oportunidade concedida, confiança e orientação acadêmica, no desenvolvimento deste trabalho.

Agradeço à professora Marli pelas sugestões de estudo, orientações e contribuições enriquecedoras durante a minha formação.

Agradeço ao colega Roddy pelo auxílio imprescindível na obtenção dos resultados finais deste trabalho.

Agradeço aos amigos Gabriel, Amarildo e Carlos Henrique pelas contribuições e sugestões finais deste trabalho.

Agradeço à CAPES, pelo apoio financeiro através da bolsa de mestrado, indispensável na realização deste trabalho.

Agradeço ao Departamento de Microonda e Óptica da FEEC e ao Laboratório LEMAC pela infra-estrutura disponibilizada.

Sem citar mais nomes, para não ser injusto, agradeço a todos os amigos do DMO, LEMAC, LCA e das disciplinas que de alguma forma contribuíram neste trabalho, proporcionando um ambiente de críticas e sugestões enriquecedoras.

Nesta oportunidade, gostaria de agradecer aos meus pais Chang e Eliza e à minha irmã Lígia, pelo apoio que proporcionaram na minha jornada acadêmica.

Cindindo a vastidão do Azul profundo,
Sulcando o espaço, devassando a terra,
A aeronave que um mistério encerra
Vai pelo espaço acompanhando o mundo.

E na esteira sem fim da azúlea esfera
Ei-la embalada n'amplidão dos ares,
Fitando o abismo sepulcral dos mares,
Vencendo o azul que ante si s'erguera.

Voa, se eleva em busca do infinito,
É como um despertar de estranho mito,
Auroreando a humana consciência.

Cheia da luz do cintilar de um astro,
Deixa ver na fulgência do seu rastro
A trajetória augusta da Ciência.

Augusto dos Anjos – A Aeronave.

Sumário

1	Introdução.....	1
1.1	Introdução às Malhas	2
1.2	Motivação e Objetivo	5
1.3	Organização do Trabalho	6
2	Malhas Bidimensionais	8
2.1	Malhas Não Estruturadas	8
2.2	Propriedades Desejadas de Elementos Finitos	11
2.3	Função de Densidade	14
3	Técnicas de Refinamento Adaptativo	16
3.1	Técnicas de Subdivisão.....	16
3.2	Triangulação de Delaunay.....	20
3.3	Algoritmos de Delaunay	26
3.4	Predicados Geométricos	28
3.5	Aspectos Qualitativos de uma Malha	30
3.5.1	Razão entre Raio Circunscrito e Menor Aresta	30
3.5.2	Razão entre Raios Inscrito e Circunscrito	33
3.5.3	Outros Critérios de Qualidade	34
3.6	Técnicas de Suavização.....	37
4	Implementação Computacional	40
4.1	Algoritmo de Subdivisão dos Elementos	40
4.2	Descrição do Método da Decomposição da Topologia	47
4.3	Integridade das Interfaces Geométricas.....	52
4.3.1	Interface Retangular.....	52
4.3.2	Interface Circular.....	54
4.3.3	Interface de Curvas Arbitrárias.....	56
4.3.4	Exemplos de Estruturas	62
5	Estudos Comparativos.....	66
5.1	Comparação entre Técnicas de Suavização	66
5.2	Comparação de Estruturas.....	71
5.2.1	Refinamento de um Guia Retangular	71
5.2.2	Refinamento de um Guia Tipo Microfita	75
5.2.3	Refinamento de um Guia Retangular Parcialmente Preenchido ..	79
6	Testes e Simulações	82
6.1	Recursos Utilizados	82
6.2	Refinamento de um Guia Óptico Retangular	83
6.3	Refinamento de um Guia Tipo Costela ('Rib')	88

6.4	Refinamento de um Guia do Tipo Enterrado	92
7	Conclusão e Trabalhos Futuros	97
7.1	Conclusões.....	97
7.2	Trabalhos Futuros	98
	Referências Bibliográficas	99
A	Método dos Elementos Finitos	105
A.1	Método dos Elementos Finitos Aplicado à Resolução da Equação de Poisson	105
A.2	Caso Bidimensional.....	107
A.3	Matriz Global	109
B	Simplexos e Padrões de Numeração	111
B.1	Simplexos.....	111
B.2	Padrões de Numeração.....	112
C	Descrição dos algoritmos	115
C.1	Algoritmo de Busca de Vizinhos Adjacentes	115
C.2	Descrição dos Algoritmos de Integridade	117
C.2.1	Algoritmo de Integridade de Interfaces Retangulares	118
C.2.2	Algoritmo de Integridade de Interfaces Circulares.....	119
C.2.3	Algoritmo de Integridade de Curvas	122
C.3	Eficiência do Programa de Elementos Finitos	124
D	Saída de Arquivos.....	127
D.1	Simulação no Aníbal.....	127
D.2	Visualização da Malha.....	129

Listas de Figuras

Fig. 1.1: Tipos de malhas de (a) elementos triangulares e (b) elementos retangulares.	3
Fig. 1.2: Características fundamentais de uma malha, partindo-se de uma (a) geometria de entrada; após a geração de malha, exemplos de (b) superposição de elementos, (c) degradação da interface, e (d) malha de bons aspectos para refinamento denso.....	4
Fig. 2.1: Malha não estruturada com a identificação dos nós.....	9
Fig. 2.2: Exemplos de malhas estruturadas de (a) domínio regular e de (b) domínio irregular; e as (c) malhas não estruturadas.	11
Fig. 2.3: (a) Geração de uma malha com (b) elementos conformes e (c) elementos não conformes.	12
Fig. 2.4: (a) Geração de uma malha com (b) elementos degenerados.	13
Fig. 2.5: Transição abrupta entre elementos.	13
Fig. 3.1: Aplicação da (a) técnica baricêntrica para subdivisão de um elemento interno; e uso da técnica da bissetção para (b) subdivisão em interface retilínea, (c) subdivisão em interface curva, e (d) subdivisão de interface interna retilínea.	17
Fig. 3.2: O processo iterativo da técnica baricêntrica na subdivisão do (a) elemento interno após duas iterações, e do (b) elemento pertencente à interface, após duas iterações.	18
Fig. 3.3: Uso da técnica da bissetção generalizada em uma (a) malha inicial, e o seu (b) refinamento denso localizado no centro de um guia retangular.	19
Fig. 3.4: Subdivisão de um elemento em quatro através das técnicas da (a) bissetção dos três lados, e dos (b) pontos médios das três arestas.	19
Fig. 3.5: (a) Disposição inicial de um circuncírculo da aresta e , com um vértice interno a ele, e (b) aumentando-se o raio do mesmo, até que se encontre um (b) circuncírculo vazio.	21

Fig. 3.6: Exemplo de uma (a) triangulação de Delaunay, e quando um (b) triângulo não é Delaunay.....	22
Fig. 3.7: Troca de uma (a) aresta não Delaunay e , pela (b) aresta de Delaunay e'	22
Fig. 3.8: Exemplo de uma triangulação de (a) Delaunay, e (b) não Delaunay ocorrendo a troca de arestas.....	23
Fig. 3.9: (a) Antes da troca de arestas, e (b) após, com a maximização dos ângulos mínimos internos.....	24
Fig. 3.10: (a) Geração de uma malha inicial côncava; a (b) checagem de um vértice interno a t_1 ; e a (c) degradação da interface.	25
Fig. 3.11: Recuperação do elemento (a) t_2 de baixa qualidade, após a (b) adição de vértices na sua interface.....	25
Fig. 3.12: Algoritmo de Bowyer/Watson, na (a) inserção de um novo ponto p' ; na verificação dos (b) triângulos, cujo circuncírculo contenham p' ; e onde as (c) arestas criadas se interceptam em p'	26
Fig. 3.13: Algoritmo de inserção incremental de Lawson, na (a) inserção de um novo ponto p' ; na (b) formação de novos elementos; e na (c) - (e) troca de arestas.....	27
Fig. 3.14: (a) Relação da razão entre o raio circunscrito e a menor aresta através de α_{\min} , e o (b) gráfico de variação de $\frac{R}{e}$ em função de α_{\min}	31
Fig. 3.15: Triângulo p_i, p_j, p_k , com o deslocamento do vértice $p_k(y)$	35
Fig. 3.16: Gráfico de variação de $p_k(y)$ em função de $\alpha, \frac{r}{R}$ e β	36
Fig. 3.17: Equilíbrio de um nó local.	38
Fig. 4.1: Fluxograma de subdivisão dos elementos.	42
Fig. 4.2: Função gaussiana pré-estabelecida, gerada (a) no Matlab, e no (b) gerador deste projeto.	43
Fig. 4.3: Função retangular pré-estabelecida, gerada (a) no Matlab, e no (b) gerador deste projeto.	44

Fig. 4.4: Soma de funções gaussianas, geradas (a) no Matlab, e (b) no gerador deste projeto, ainda com a definição de 4 interfaces internas e retangulares.....	44
Fig. 4.5: Variação da posição do <i>spot</i> em ordem seqüencial.	46
Fig. 4.6: (a) Refinamento de uma malha inicial, aplicando-se a técnica baricêntrica, e após isso, (b) aplicando-se a troca de arestas em $\overline{p_i p_j}$ e, depois, em $\overline{p_i p_k}$	48
Fig. 4.7: (a) Aplicação da bissecção pela maior aresta, seguida pela (b) troca de arestas em $\overline{p_i p_m}$	49
Fig. 4.8: Aplicação da troca de arestas em $\overline{p_i p_n}$	50
Fig. 4.9: Malha final obtida e o hitórico de todos os elementos gerados.....	51
Fig. 4.10: A localização do novo nó $p' = (x', y')$ num (a) elemento de interface externa, e num (b) elemento de interface interna.....	53
Fig. 4.11: Exemplo de geração de malha para uma geometria em (a) formato “L”, e em (b) formato “H”.....	54
Fig. 4.12: Integridade da interface circular, a partir de uma (a) disposição inicial, e após a (b) inserção do ponto de recuperação.....	55
Fig. 4.13: Exemplo de geração de malha, com (a) 818 elementos, e (b) 4632 elementos.....	56
Fig. 4.14: Definição de uma curva arbitrária através dos (a) pontos de controle P_0, \dots, P_5 , e a (b) geração dos nós p_0, \dots, p_9 da curva.	58
Fig. 4.15: Curva arbitrária, com (a) variação de 0,25 no parâmetro t ; o (b) elemento superior com parâmetros t_{\min} , t_{med} e t_{\max} ; a (c) inserção de p' sobre a curva.....	59
Fig. 4.16: Estrutura com uma (a) curva arbitrária simétrica interna na região central; a (b) definição dos pontos de controle, e posteriormente, a geração de nós da curva com variação de $t = 0,5$ no parâmetro.	59
Fig. 4.17: Geração de uma malha com uma curva simétrica, a partir de uma (a) malha inicial; a (b) divisão em 2 intervalos, e o (c) processo iterativo de quebra.....	60
Fig. 4.18: (a) Curva arbitrária assimétrica com a definição dos pontos de controle; e o (b) processo iterativo de quebra.....	60

Fig. 4.19: Geração de malha partindo-se de uma (a) estrutura de curva arbitrária fechada, e os (b) nós gerados pertencentes à mesma com a definição de quatro intervalos. A partir dos intervalos definidos, monta-se uma (c) malha inicial para o refinamento.	61
4.20: Resultados de uma malha refinada com (a) 660 elementos, e com (b) 1190 elementos.....	62
Fig. 4.21: Exemplos de geração de malhas.	63
Fig. 4.22: Exemplos de geração de malhas com interfaces circulares.....	64
Fig. 4.23: Geração de malhas em estruturas complexas.	65
Fig. 5.1: Malha obtida após a (a) técnica de Delaunay, aplicando-se a suavização laplaciana com (b) uma iteração, (c) duas iterações, e (d) três iterações.	67
Fig. 5.2: (a) Malha obtida após a técnica de Delaunay, e o seu (b) histograma do fator de qualidade em função do número de elementos normalizados.	68
Fig. 5.3: (a) Malha obtida com a aplicação da suavização laplaciana após a técnica de Delaunay, e o seu (b) histograma do fator de qualidade em função do número de elementos normalizados.	69
Fig. 5.4: (a) Malha obtida com a aplicação da suavização laplaciana restrita após a técnica de Delaunay, e o seu (b) histograma do fator de qualidade em função do número de elementos normalizados.	70
Fig. 5.5: (a) Esboço de um guia óptico retangular e a sua (b) malha inicial.	71
Fig. 5.6: Guia óptico retangular e sua (a) geração de malha numa estrutura completa, e ao lado, o (b) detalhe da região do canal interno, aumentado em seis vezes.....	72
Fig. 5.7: Gráficos de distribuição. (a) histograma de qualidade, (b) histograma de ângulos internos.....	73
Fig. 5.8: Gráficos de distribuição do número de elementos normalizados em função do fator e qualidade alpha para cinco densidades de refinamento.....	74
Fig. 5.9: Gráfico do tempo de execução.....	75
Fig. 5.10: Estrutura de um guia tipo microfita.	76
Fig. 5.11: Geração de malha sem a cavidade delimitada.....	77
Fig. 5.12: Geração de malha com a cavidade delimitada.....	77

Fig. 5.13: Adição de vértices na divisão central.	77
Fig. 5.14: Adaptação da (a) malha através da adição de novos vértices na interface, e a sua (b) ampliação em três vezes.	78
Fig. 5.15: Distribuição do (a) fator de qualidade e dos (b) ângulos internos em função do número de elementos normalizados.	79
Fig. 5.16: Esboço de estrutura de guia retangular parcial.	79
Fig. 5.17: Refinamento de guia retangular parcial. (a) Refinamento produzido (b) Refinamento modificado.	80
Fig. 5.18: Distribuição do (a) fator de qualidade e dos (b) ângulos internos em função do número de elementos normalizados, da Figura 5.17 (a).	80
Fig. 5.19: Distribuição do (a) fator de qualidade e dos (b) ângulos internos em função do número de elementos normalizados, da Figura 5.17 (b).	81
Fig. 6.1: Definindo a (a) proporção do guia óptico retangular, as (b) coordenadas desejadas, e os (c) tipos de materiais.	84
Fig. 6.2: Geração de malhas de um guia óptico retangular e as suas seguintes identificações de saída: (a) geom1_dens1, (b) geom1_dens2, (c) geom1_dens3, (d) geom1_dens4, e (e) geom1_dens5.	85
Fig. 6.3: Gráficos do (a) $\log(\text{erro})$ em função de λ , e da (b) queda linear do $\log(\text{erro})$ em função do número de elementos, do guia retangular.	87
Fig. 6.4: Definindo a (a) proporção do guia tipo costela, as (b) coordenadas desejadas, e os (c) tipos de materiais.	89
Fig. 6.5: Geração de malhas de um guia tipo costela e as suas seguintes identificações de saída: (a) geom2_dens1, (b) geom2_dens2, (c) geom2_dens3, (d) geom2_dens4, e (e) geom2_dens5.	90
Fig. 6.6: (a) Gráfico do $\log(\text{erro})$ em função de λ , (b) Gráfico de queda linear do $\log(\text{erro})$ em função do número de elementos, do guia tipo costela.	91
Fig. 6.7: Definindo a (a) proporção do guia tipo enterrado, as (b) coordenadas desejadas, e os (c) tipos de materiais.	93
Fig. 6.8: Geração de malhas de um guia tipo costela e as suas seguintes identificações de saída: (a) geom3_dens1, (b) geom3_dens2, (c) geom3_dens3, (d) geom3_dens4, e (e) geom3_dens5.	94

Fig. 6.9: Identificação dos materiais, da malha da Figura 6.8 (a), com zoom em 12x da estrutura interna.....	95
Fig. 6.10: (a) Gráfico de $\log(erro)$ em função de λ , (b) Gráfico de queda linear de $\log(erro)$ em função do número de elementos, do guia tipo enterrado.	96
Fig. A.1: Elemento triangular linear.	108
Fig. A.2: Elemento triangular linear. (a) Exemplo arbitrário de malha 2-D, (b) Padrão de numeração local dos nós dos elementos.	109
Fig. B.1: Simplexos para (a) $n = 1$, (b) $n = 2$ e (c) $n = 3$	111
Fig. B.2: Padrão de numeração dos nós globais, (a) antes, com quatro nós de identificação, e (b) após a técnica baricêntrica, produzindo-se seis nós.	112
Fig. B.3: Padrão de numeração dos nós locais. (a) antes, com a numeração dos elementos iniciais, e (b) após a técnica baricêntrica.	113
Fig. B.4: Numeração dos elementos, (a) antes, com malha inicial de dois elementos, e (b) após, produzindo-se seis elementos.	113
Fig. C.1: (a) Malha de entrada. (b) Mapeamento de elementos ligados ao ponto.	116
Fig. C.2: Detecção de vizinho através da intersecção dos elementos ligados aos nós 1 e 5.	116
Fig. C.3: Disposição da numeração local em elementos de interfaces distintas.	117
Fig. C.4: Possíveis disposições do elemento na interface. (a) Elemento interno ao círculo, e (b) elemento externo.	120
Fig. C.5: Exemplo de diagrama de classes.	125
Fig. C.6: Exemplo simples de diagrama ontológico.	125
Fig. D.1: Visualização no GiD.	130

Lista de Tabelas

Tab. 2.1: Matriz de conectividade dos elementos da Figura 2.1.	10
Tab. 3.1: Algoritmo de teste de nó dentro do circuncírculo.	29
Tab. 3.2: Tabela de cálculo de B , α_{\min} e α_{\max}	32
Tab. 3.3: Tabela de cálculo do fator de qualidade normalizado entre α , $\frac{r}{R}$ e β	35
Tab. 3.4: Algoritmo de suavização laplaciana.	38
Tab. 3.5: Algoritmo de suavização laplaciana restrita.	39
Tab. 4.1: Algoritmo de subdivisão dos elementos.	41
Tab. 5.1: Tabela de crescimento da porcentagem de elementos na faixa de $0,9 \leq \alpha \leq 1,0$	74
Tab. 6.1: Características do refinamento de um guia óptico retangular.	85
Tab. 6.2: Comprimento de onda e o cálculo do índice modal, do guia retangular.	86
Tab. 6.3: Cálculo do $\log(erro)$, do guia retangular, em referência ao GiD.	86
Tab. 6.4: Características do refinamento de um guia tipo costela.....	89
Tab. 6.5: Comprimento de onda e o cálculo do índice modal, do guia tipo costela.	91
Tab. 6.6: Cálculo do $\log(erro)$, do guia tipo costela, em referência ao GiD.	91
Tab. 6.7: Características de refinamento de um guia tipo enterrado.....	93
Tab. 6.8: Comprimento de onda e o cálculo do índice modal, do guia enterrado.	95
Tab. 6.9: Cálculo de $\log(erro)$, do guia tipo enterrado, em referência ao GiD.	96
Tab. A.1: Matriz que associa o elemento com sua numeração local e global. ..	109
Tab. A.2: Matriz global, com a inclusão da primeira matriz elementar.	110
Tab. A.3: Matriz global resultante.	110

Tab. C.1: (a) Malha de entrada. (b) Mapeamento de elementos ligados ao ponto.	116
Tab. C.2: Algoritmo de integridade de bordas retangulares.	118
Tab. C.3: Algoritmo de integridade de bordas circulares.	121
Tab. C.4: Algoritmo de integridade de interface curva.	123
Tab. D.1: Exemplo de saída de arquivo do gerador de malha para simulação.	128
Tab. D.2: Exemplo de saída de arquivo para visualização.	129

Lista de Acrônimos

MEF: Método dos Elementos Finitos.....	1
CAD: Computer Aided Design.....	1
DMO: Departamento de Microonda e Óptica	96
LEPP: Longest Edge Path Propagation.	96
JAI: Java Advanced Imaging	96

Capítulo 1

Introdução

O Método dos Elementos Finitos (MEF), descrito no Apêndice A, é uma técnica que possibilita simular o comportamento de um objeto através de uma geometria previamente definida. Essa geometria deve possuir uma determinada estrutura e material que poderá ser utilizada para diversos testes como térmico, mecânico, eletromagnético, etc, a fim de prever seu comportamento real.

Esse método foi introduzido, em 1909, por Walter Ritz, para a solução aproximada de problemas em mecânica dos sólidos deformáveis. A associação de elementos finitos com ferramentas de desenho auxiliado por computador (CAD – *Computer Aided Design*) [1] teve seu início a partir da década de 50, em projetos de aeronaves. Nessa época, esse ramo industrial era um dos únicos capazes de suportar altos custos de *hardware* e *software* para o trabalho. Atualmente, com a evolução da informática e o barateamento de equipamentos computacionais, os elementos finitos têm sido utilizados como uma técnica adequada para a simulação computacional de problemas complexos em diferentes campos da engenharia: na mecânica, na aerodinâmica, e o seu uso na engenharia elétrica tem o objetivo de auxiliar no desenvolvimento de projetos, como as fibras ópticas. De fato, a complexidade destes dispositivos em estudo, nem sempre permite obter resultados através de métodos analíticos, sendo necessário o uso de métodos numéricos que produzam soluções aproximadas, como é o caso dos elementos finitos.

Essas aproximações são obtidas através de simulações, que podem ser interpretadas como modelagens de objetos específicos reais ou imaginários de sistemas. Elas permitem ao projetista o conhecimento prévio a respeito das características do objeto em estudo, por meio da simulação virtual, sem a necessidade de gasto de material e de tempo, que são comuns à construção de um protótipo real.

De forma geral, as simulações computacionais agregadas a um sistema FEM possuem os seguintes procedimentos:

- Pré-Processamento: fornece a estrutura a ser analisada, bem como a característica do material de que é composto e a produção do domínio subdividido em partes menores, denominados elementos finitos.
- Processamento: efetua o cálculo numérico propriamente dito, utilizando-se os dados gerados na primeira etapa do processo na construção das matrizes elementares e montagem da matriz global.
- Pós-processamento: apresenta os resultados da forma mais eficiente possível de modo a facilitar a sua análise.

A fase de pré-processamento, que será vista ao longo deste trabalho, está associada à geração de malhas. A inexistência desta etapa dificulta bastante o trabalho do pesquisador, pois a elaboração manual de uma malha refinada é uma tarefa complexa e sujeita a erros.

1.1 Introdução às Malhas

Existem diversas aplicações para as malhas, sendo que seu uso se destaca bastante em métodos numéricos (solução de equações diferenciais), interpolação (amostragem de dados) e computação gráfica (visualização de geometrias 2D e 3D).

Segundo Baude [2], que descreve uma nova técnica de refinamento adaptativo de malhas 2D e 3D para a solução numérica de problemas eletromagnéticos, a divisão de um domínio do espaço em elementos geométricos é denominada malha. Esta pode ser composta por diversas formas de elementos como triângulos, quadrados e círculos. Entretanto, costuma-se utilizar um mesmo tipo de elemento para simulações computacionais. Os elementos mais comuns de representação são os triângulos e quadriláteros. O primeiro, utilizado neste projeto, é de natureza mais simples por ser constituído de três (linear) ou seis (quadrático) nós. A Figura 1.1 detalha os componentes de uma malha. Cada triângulo ou retângulo corresponde a um elemento e cada vértice de um deles é denominado nó ou ponto nodal.

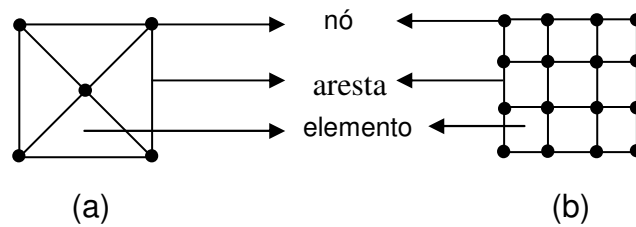


Fig. 1.1: Tipos de malhas de (a) elementos triangulares e (b) elementos retangulares.

O trabalho de Lisboa [3], que descreve a geração de malhas para geometrias curvas, apresenta algumas características fundamentais, a respeito de uma malha geométrica, que devem ser seguidas. A Figura 1.2 (a) mostra uma estrutura definida, enquanto as figuras (b), (c) e (d) são exemplos de geração de malhas a fim de ilustrar as características descritas a seguir:

- Não deve existir sobreposição de elementos. Na Figura 1.2 (b), a impressão simultânea de dois elementos provoca a inexistência de um novo tipo de triângulo, visto através das linhas pontilhadas. Portanto, elementos adjacentes devem se encostar apenas através de suas arestas, as quais são representadas de forma única. Além disso, não devem existir dois nós com a mesma coordenada;

- O aspecto representativo da geometria de entrada da Figura 1.2 (a) não deveria sofrer alterações, resultando na Figura 1.2 (c). Os subdomínios deveriam se adaptar a esta representação; no entanto, na Figura 1.2 (c) observa-se a degradação do contorno do domínio;
- Possibilidade de inserção de elementos onde se necessita. A partir da malha inicial da Figura 1.2 (d), deve ser possível obter maior refinamento nas regiões que se deseja;
- Deformação da parte da malha, de modo a poder incrementar ou retirar parte da interface geométrica da malha gerada para problemas de análise de sensibilidade;
- Aspectos qualitativos de um elemento que estão descritos especificamente nas Seções 2.2 e 3.5;

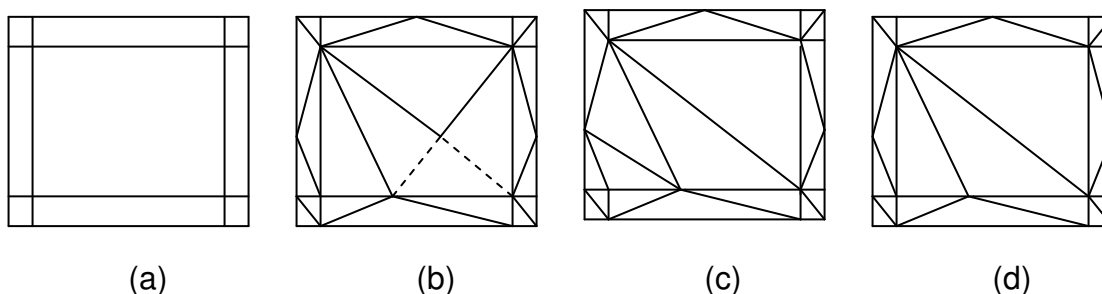


Fig. 1.2: Características fundamentais de uma malha, partindo-se de uma (a) geometria de entrada; após a geração de malha, exemplos de (b) superposição de elementos, (c) degradação da interface, e (d) malha de bons aspectos para refinamento denso.

Recentemente, novos conceitos em computação gráfica impulsionaram o surgimento de novas estratégias na geração de malhas de elementos finitos. A área de interesse deste trabalho, voltada para a computação gráfica, conduz os resultados práticos direcionados ao eletromagnetismo, adaptando as técnicas de refinamento desenvolvidas no laboratório e solucionando os problemas numéricos em questão.

1.2 Motivação e Objetivo

A linha de pesquisa de computação gráfica do Departamento de Microonda e Óptica (DMO) possui um histórico pouco extenso no desenvolvimento de bibliotecas computacionais gráficas. Dentre elas, pode-se mencionar os algoritmos de pré-processadores gráficos bidimensionais e tridimensionais para análise de elementos finitos. Faz-se aqui um breve histórico na área de geração de malhas. H. E. Figueroa é autor de uma ferramenta, desenvolvida em 1990, denominada HMesh, que aborda a geração de malhas de elementos finitos bidimensionais através da técnica da bissecção generalizada baseada em interfaces retangulares. Em 2000, Baude [2] propôs um novo algoritmo de geração de malhas através da conjuntura da técnica de bissecção com o caminho de propagação da maior aresta (LEPP – *Longest Edge Path Propagation*). Atualmente, Malavazi [4] descreve um ambiente computacional gráfico interativo para geração de modelos de elementos finitos 2-D e 3-D utilizando a programação orientada a objeto.

As rotinas e funções para a geração de malhas desenvolvidas no departamento são frutos de programação estruturada, o que as torna dependentes do pré-processador. Dessa forma, no enfoque de desenvolvimento de programas complexos, é imprescindível organizar o código a fim de permitir versões sucessivas, manutenção, e reuso de código. Além disso, as bibliotecas existentes, tanto o HMesh quanto o sistema desenvolvido por Baude [2], possuem limitações na entrada de dados quando se tem geometria de interface mais complexa, como por exemplo, curvas arbitrárias 2-D.

O objetivo deste trabalho é o desenvolvimento de algoritmos para geração de malhas adaptativas de elementos triangulares, com a finalidade de estudar a propagação de ondas eletromagnéticas em domínios bidimensionais. A implementação se dá através de diversas técnicas de refinamento, dentre elas, o método da troca de arestas denominado Delaunay e a técnica de suavização de

Laplace aplicada à malha. Este estudo deverá explorar a geração de malha em diversos domínios, tais como retangular, circular e curva arbitrária. Pretende-se alcançar resultados satisfatórios através da utilização de linguagem de programação orientada a objeto, onde a obtenção de códigos robustos que atenda aos critérios de qualidade geométrica e capacidade de adaptação em estruturas complexas é parte deste estudo. Como fruto deste trabalho, a implementação permitirá favorecer a integração com as bibliotecas computacionais existentes e que serão desenvolvidas no DMO. Além disso, a possibilidade de torná-lo funcionalmente comparável a um sistema comercial, torna relevante a necessidade de componentes orientados a objeto.

1.3 Organização do Trabalho

Este trabalho está organizado em 4 partes: a introdução, vista no Capítulo 1; os conceitos fundamentais, compostos pelos Capítulos 2 e 3; a implementação computacional e a análise de resultados formados pelos Capítulos 4, 5 e 6; e a conclusão, no Capítulo 7.

O Capítulo 1 apresenta uma idéia geral do conceito e das características fundamentais na geração de malhas.

No Capítulo 2 é apresentada uma breve discussão teórica a respeito de malhas não estruturadas bidimensionais, bem como as propriedades gerais desejadas dos elementos que as compõem e o critério utilizado para provocar a subdivisão deles.

Descreve-se, no Capítulo 3, as técnicas de refinamento, como, por exemplo, o algoritmo de Delaunay. Também são descritos, de forma detalhada, os critérios qualitativos na avaliação de uma malha. E finalmente, um estudo a respeito da técnica da suavização Laplaciana e a sua variação restrita.

A implementação tem início a partir do Capítulo 4, onde é apresentada uma descrição da metodologia de decomposição de uma malha inicial, e também as

soluções para geração de malhas em diversos domínios utilizando-se os algoritmos de integridade de bordas.

O Capítulo 5 mostra um estudo comparativo dos resultados obtidos com literaturas vistas nas teses de doutorado de Baude [2] e Ribeiro Filho [5].

No Capítulo 6 analisam-se os resultados obtidos através de testes e simulações a partir de estrutura de guias ópticos de dimensões reais.

O Capítulo 7 apresenta as conclusões deste trabalho, e as propostas de trabalhos futuros.

Capítulo 2

Malhas Bidimensionais

Este trabalho tem, como ponto de partida, o estudo de conceitos a respeito de malhas bidimensionais não estruturadas. Essas malhas são compostas pelas seguintes informações: nós globais, nós locais e a identificação dos elementos. Esses dados formam uma tabela de registro denominada matriz de conectividade. Após isso, serão descritas as propriedades desejadas de elementos finitos, quanto aos seus aspectos qualitativos. Em seguida, na Seção 2.2, será abordada a função de densidade que é responsável pelo critério de subdivisão dos elementos que compõem a malha.

2.1 Malhas Não Estruturadas

Segundo Ribeiro Filho [5] e Batista [6], cujos trabalhos relacionam-se com a geração automática e adaptativa de malhas para elementos finitos 2-D, o método dos elementos finitos utiliza células discretas generalizadas chamadas de malhas não estruturadas. Assim, para cada ponto particular, a conexão com outros nós deve ser definida explicitamente em uma matriz de conectividade. Essa matriz é importante para localização de um vértice específico numa malha; fornecer dados para a próxima iteração do algoritmo, para posterior atualização dos elementos; servir como arquivo de saída compatível com outros programas existentes; e

possibilitar a obtenção de informações como o número de nós e suas coordenadas, a quantidade total de elementos e seus materiais.

Para se entender a matriz de conectividade, acrescenta-se que cada triângulo é formado por três numerações locais de 1 a 3. Esta numeração é uma identificação específica ordenada no sentido anti-horário e está associada a um dos três vértices de um elemento. Existe também a numeração global, que são os nós de identificação geral da malha. Assim, é possível localizar um ponto nodal de duas formas: pela numeração global ou pela numeração local associada ao número do elemento. Informações mais detalhadas a respeito do modo que se efetua a numeração dos elementos, podem ser vistas no Apêndice B.

Na Figura 2.1, tem-se um exemplo de malha cuja numeração global vai de 1 a 24. Na Tabela 2.1, cria-se um mapeamento de todos os triângulos da malha da Figura 2.1, onde existem 32 elementos. Como exemplo, tem-se o elemento 18, que é formado pelos nós 11, 17 e 14. Supondo que essa seqüência seja, respectivamente, a ordem da numeração local de 1 a 3, então tem-se que o elemento 18, com numeração local 3, corresponde ao nó global 14.

A distinção entre a numeração local e global está no fato da primeira ter, como objeto de referência, um elemento, enquanto o objeto de referência da segunda é um ponto. Essa distinção, na prática de programação, é importante, pois um ponto carrega características de coordenadas num espaço, enquanto no elemento, são os nós que o compõe.

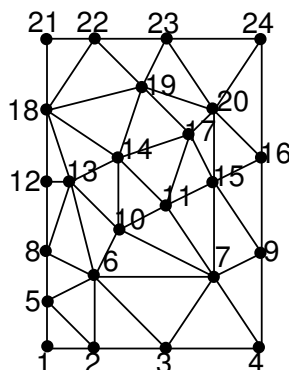


Fig. 2.1: Malha não estruturada com a identificação dos nós.

El.	Nós	El.	Nós	El.	Nós
1	(1, 2, 5)	12	(7, 15, 11)	23	(14, 17, 19)
2	(2, 6, 5)	13	(7, 9, 15)	24	(15, 20, 17)
3	(2, 3, 6)	14	(8, 13, 12)	25	(15, 16, 20)
4	(3, 7, 6)	15	(9, 16, 15)	26	(16, 24, 20)
5	(3, 4, 7)	16	(10, 14, 13)	27	(17, 20, 19)
6	(4, 9, 7)	17	(10, 11, 14)	28	(18, 22, 21)
7	(5, 6, 8)	18	(11, 17, 14)	29	(18, 19, 22)
8	(6, 13, 8)	19	(11, 15, 17)	30	(19, 23, 22)
9	(6, 10, 13)	20	(12, 13, 18)	31	(19, 20, 23)
10	(6, 7, 10)	21	(13, 14, 18)	32	(20, 24, 23)
11	(7, 11, 10)	22	(14, 19, 18)		

Tab. 2.1: Matriz de conectividade dos elementos da Figura 2.1.

Diferentemente das malhas não estruturadas observadas nas Figuras 2.1 e 2.2 (c), tem-se também as chamadas estruturadas, em que estas utilizam grades cartesianas retangulares, conforme mostradas nas Figuras 2.2 (a) e (b). As malhas estruturadas consistem em um conjunto de coordenadas que mapeiam as conectividades dos elementos em uma matriz de forma mais simples. Assim, conforme visto na Figura 2.2 (a), num ponto (i, j) , é fácil observar que os vértices adjacentes são $(i+1, j)$ e $(i, j+1)$, onde i e j representa a identificação de uma matriz de pontos. Na prática, o modo que essa malha se comporta torna a criação de algoritmos e softwares altamente eficientes. Entretanto, de acordo com Plassmann [7], essas estruturas não são utilizadas em domínios mais complexos, ou seja, estruturas como as mostradas na Figura 2.2 (c), pois não é sempre possível organizar os elementos de maneira que se adaptem bem ao domínio. Neste caso, deve-se optar por malhas não estruturadas.

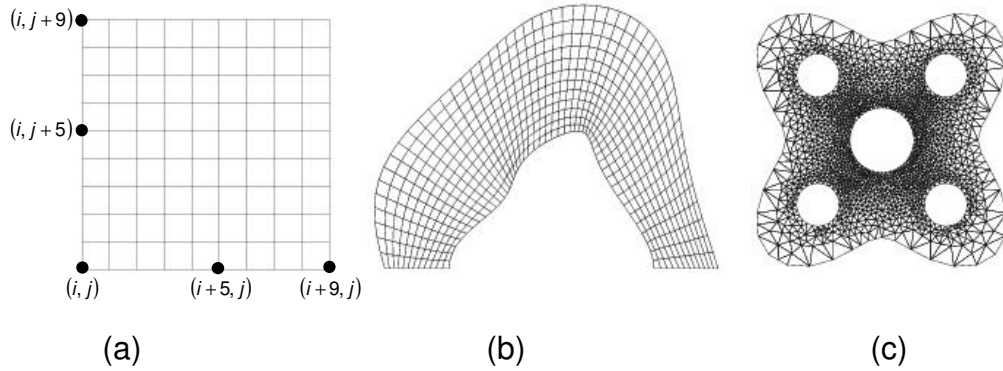


Fig. 2.2: Exemplos de malhas estruturadas de (a) domínio regular e de (b) domínio irregular; e as (c) malhas não estruturadas.

O desenvolvimento de algoritmos adaptativos de malhas não estruturadas aplicado ao MEF tem-se destacado em diversos trabalhos, que podem ser obtidos em: Bryant [8], Shephard [9], Karamete [10] e Nambiar [11]. Finalmente, é importante salientar que a construção dos elementos deve seguir algumas propriedades qualitativas desejadas, a fim de se evitar erros nos resultados numéricos.

2.2 Propriedades Desejadas de Elementos Finitos

A necessidade de se obter uma discretização coerente que gere uma malha computacionalmente eficaz impõe certos fatores de qualidade à malha, que precisam ser satisfeitos. Segundo Baude [2], os elementos que formam a malha, primeiramente, devem condizer com a conformidade. Basicamente, esta é obtida quando um ponto interno ao domínio divide uma aresta de um elemento, seja ou não da interface, e esse ponto também divide o seu elemento adjacente que compartilha a mesma aresta. Para exemplificar esta propriedade, considera-se inicialmente a geração de uma malha, mostrada na Figura 2.3 (a), onde as Figuras 2.3 (b) e (c) são partes desmembradas desta malha e a linha tracejada representa uma interface. Na Figura 2.3 (b), observa-se que cada aresta de um elemento faz fronteira com apenas um elemento adjacente. Diferentemente, na Figura 2.3 (c), a

aresta dos elementos 6 e 7 é apenas parte da aresta do elemento 8. Ainda na Figura 2.3 (c), os elementos 9 e 10, que fazem parte da interface, possuem o elemento 11 como vizinho comum, provocando uma não conformidade. A interface, que indica a delimitação de dois ou mais tipos de materiais na malha, em consequência disso, também sofre degradação.

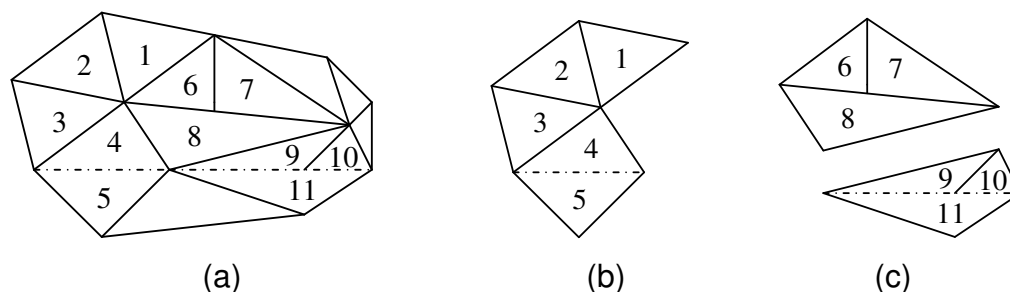


Fig. 2.3: (a) Geração de uma malha com (b) elementos conformes e (c) elementos não conformes.

Também dentro deste contexto de qualidade, surge uma outra propriedade na busca de se evitar os chamados elementos degenerados. De acordo com Shewchuk [12], que descreve a geração de malhas utilizando a técnica de Delaunay, as degenerações dos elementos diminuem a qualidade numérica da solução obtida através da simulação. Um resultado numérico tende a ser coerente quando a maior parte dos elementos de uma malha tenha um fator de qualidade aceitável. Assim, as simulações exigem que esses elementos sejam regulares, ou seja, mais próximos de triângulos eqüiláteros.

Dessa maneira, o aspecto qualitativo de um elemento se resume na avaliação do mesmo dentro de uma escala normalizada. Essa escala concentra valores de zero a um que classifica, respectivamente, um elemento de pior à melhor qualidade. Na geração de malha, é necessário evitar valores próximos de zero, pois estão associados aos elementos degenerados. O cálculo do fator de qualidade de um elemento encontra-se descrito detalhadamente no Capítulo 3. A Figura 2.4 (a) mostra uma malha e a Figura 2.4 (b) são elementos desmembrados desta malha. Na Figura 2.4 (b) observam-se dois exemplos de elementos, cujos

fatores são classificados como de baixa qualidade, pois seus ângulos internos estão próximos de 0 ou 180 graus.

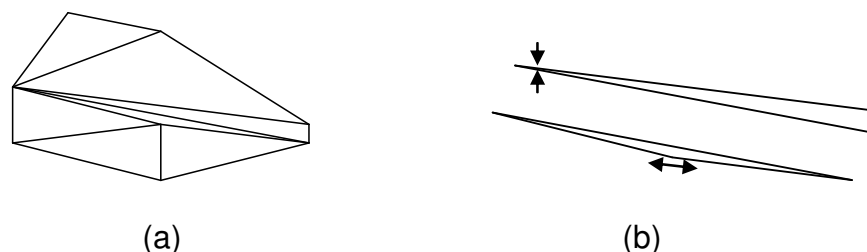


Fig. 2.4: (a) Geração de uma malha com (b) elementos degenerados.

Ademais, acrescenta-se ainda outra propriedade que é a transição entre elementos muito pequenos e grandes. Conforme mostrado na Figura 2.5, o conjunto de elementos pequenos forma um agrupamento à esquerda e, à direita, transitam abruptamente para elementos grandes. Uma malha satisfatória deve evitar isso, produzindo uma transição gradual entre eles. Portanto, é necessário gerar elementos de dimensões intermediárias ou deslocar os nós dos elementos pequenos a fim de aumentar sua dimensão, causando uma gradação.

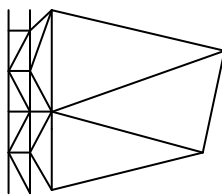


Fig. 2.5: Transição abrupta entre elementos.

Todos esses fatores são relevantes para se obter uma malha final de aspecto geométrico de alta qualidade. Na próxima seção, será abordada outra característica a respeito das malhas, que é a função de densidade. Esta, que é pré-definida pelo usuário, provocará quebras sucessivas dos elementos maiores em sub-elementos, gerando uma malha mais densa.

2.3 Função de Densidade

A função de densidade é responsável pelo monitoramento e refinamento da malha. O critério da subdivisão dos elementos finitos é determinado por uma função pré-definida. É possível utilizar diferentes formas de função que representam o problema desejado. Uma das formas mais comuns é utilizar uma função gaussiana como feixe de excitação de campo, mostrado em (2.1):

$$f(x, y) = A \cdot \exp \left\{ - \left[\frac{(x - x_0)}{d_1} \right]^2 - \left[\frac{(y - y_0)}{d_2} \right]^2 \right\} \quad (2.1)$$

A Equação 2.1 define um *spot* (região), composto pelos seguintes parâmetros de entrada: A , a amplitude; x_0 e y_0 , o centro do *spot*; d_1 e d_2 , as distâncias em relação ao centro.

Caso se some uma quantidade limitada de funções gaussianas, a geração final da malha será composta por um conjunto visível de *spots*.

Dentre várias formas, é possível definir, por exemplo, um *spot* retangular, visto em (2.2):

$$f(x, y) = A \cdot (x - a) \cdot (x - b) \cdot (y - c) \cdot (y - d), \quad (2.2)$$

onde a e b correspondem, respectivamente, à coordenada mínima e máxima dos vértices do retângulo no eixo horizontal; c e d , a mínima e a máxima no eixo vertical.

Tem-se ainda que o peso individual w_i de cada elemento é dado por (2.3), que pode ser obtido em Fernandez [13]:

$$w_i = \frac{A_{\text{elemento } i}}{3} \cdot \left[\sum_{k=1}^3 f(x_k, y_k) \right] \quad (2.3)$$

Nesta equação, x_k e y_k , para $k = 1, \dots, 3$, pertencem às coordenadas dos vértices do elemento i . Existem outras formas para o cálculo do peso individual,

como por exemplo, substituir a soma das funções de (2.3), por $[f(\bar{x}, \bar{y})]$, onde \bar{x} e \bar{y} são respectivamente,

$$\bar{x} = \frac{\sum_{k=1}^3 x_k}{3} \text{ e } \bar{y} = \frac{\sum_{k=1}^3 y_k}{3} \quad (2.4)$$

A área de cada elemento, $A_{\text{elemento } i}$, é dada por:

$$A_{\text{elemento } i} = \frac{(x_2.y_3 - x_3.y_2 + x_3.y_1 - x_1.y_3 + x_1.y_2 - x_2.y_1)}{2,0} \quad (2.5)$$

A densidade total W_m é calculada sobre o domínio do problema, da forma:

$$W_m = \sum_{i=1}^{TotalElemento} w_i, \quad (2.6)$$

onde *TotalElemento* é o número total de elemento em questão. E a partir disso, a densidade por elemento, VE , é calculado da seguinte forma:

$$VE_m = \frac{W_m}{Nelem_m} \quad (2.7)$$

Em (2.6) e (2.7), m representa a identificação da malha em questão, referente à $Malha_m$. A $Malha_m$ identifica a quantidade de iterações do algoritmo, e o estado do conjunto de elementos gerados. À medida que esses elementos passam por uma subdivisão sucessiva, cria-se uma nova identificação da malha. Essa subdivisão é determinada através de uma condição, que pode ser vista no trabalho de Nambiar [11], onde para cada elemento i , se $w_i > VE_m$, então ocorrerá a quebra do elemento i . Entretanto, é necessário atentar para a possibilidade de recursão infinita, ou seja, quando a quebra de elementos não possui uma condição para terminar. Isso pode ser solucionado através de um número de elementos estimados como parâmetro definido pelo usuário.

Portanto, este capítulo descreveu a função de densidade como a primeira etapa na decisão de refinamento de uma malha. O refinamento torna-se relevante neste momento, pois será tomado como referência no capítulo seguinte, que além de mostrar o algoritmo de subdivisão, também trata das técnicas de Delaunay.

Capítulo 3

Técnicas de Refinamento Adaptativo

Este capítulo aborda algumas técnicas de refinamento, tais como: as técnicas baricêntrica e da bissecção para a subdivisão dos elementos; tipos de algoritmos de triangulação de Delaunay como, por exemplo, o *edge-flip*, que é responsável pela troca de arestas entre elementos adjacentes; apresentação do cálculo de predicados geométricos, que está associada à técnica de Delaunay, para detectar vértices contidos em circuncírculos; a suavização de Laplace, que desloca os nós da malha permitindo uma suavização no seu aspecto; e, por fim, um estudo a respeito de cálculos de fatores de qualidade, como a razão entre raio circunscrito e menor aresta, a razão entre raios inscrito e circunscrito, entre outros.

3.1 Técnicas de Subdivisão

A técnica baricêntrica, método de refinamento implementado neste trabalho, está associada à função de densidade, que foi citada na Seção 2.3. Essa subdivisão associa-se à inserção de um novo nó no baricentro de um elemento. Os nós existentes conectam-se a esse novo nó, gerando-se três novos elementos. Quando o elemento é de categoria interna, não existe problema de conformidade, pois o ponto baricêntrico não se localiza sobre a aresta do elemento. Na Figura 3.1 (a), à esquerda, um exemplo de elemento interno; e à direita, a quebra em três elementos menores.

Entretanto, a técnica baricêntrica vista em trabalhos de Fernandez [13] e Fernandes [14], que discutem estratégias de refinamento em malhas adaptativas, também incorporam a técnica da bissecção em condições específicas. Nos casos em que o elemento possui uma aresta posicionada numa interface, como nas Figuras 3.1 (b) e (c), representada pela linha tracejada, o novo nó localizar-se-á de duas formas: no ponto médio da aresta de uma interface retilínea, mostrada na Figura 3.1 (b); e sobre o arco médio da interface curva, que define a fronteira, no caso da Figura 3.1 (c). Se o elemento possuir uma aresta posicionada numa interface interna, ou seja, entre dois elementos adjacentes, ilustrada na Figura 3.1 (d), é necessário que se divida também o elemento adjacente, a fim de se manter a conformidade da malha.

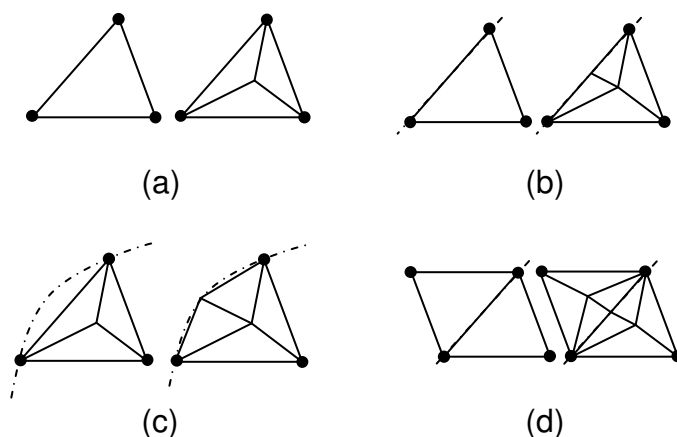


Fig. 3.1: Aplicação da (a) técnica baricêntrica para subdivisão de um elemento interno; e uso da técnica da bissecção para (b) subdivisão em interface retilínea, (c) subdivisão em interface curva, e (d) subdivisão de interface interna retilínea.

Como o processo de refinamento adaptativo é recursivo, nenhum elemento deve ser dividido mais do que uma vez a cada iteração do algoritmo, justamente para não deteriorar a qualidade da malha. Como exemplo disso, tratando-se de elementos isolados, tem-se a Figura 3.2 que mostra um exemplo de recursividade após duas iterações. Na Figura 3.2 (a) à esquerda, o centróide é inserido, gerando-se três novos elementos e na segunda iteração à direita, um novo centróide em cada elemento formando-se nove novos elementos. De forma

análoga, no caso da bissecção simples, a imagem 3.2 (b) à direita produziu seis novos elementos. Pode-se notar, na segunda iteração, o surgimento de elementos com ângulos agudos e obtusos em ambas as figuras da direita. Para resolver esse problema de degeneração, o procedimento mais adequado é realizar uma checagem na primeira iteração por elementos anômalos, e corrigir o aspecto geométrico da malha, para em seguida, aplicar a segunda iteração. Não é conveniente permitir que uma única iteração manipule subdivisões contínuas dos elementos, e somente ao final de todo o processo corrigir o problema de qualidade dos elementos, pois elementos degenerados persistirão.

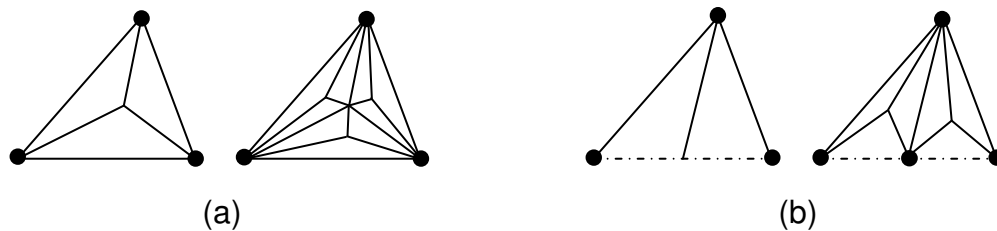


Fig. 3.2: O processo iterativo da técnica baricêntrica na subdivisão do (a) elemento interno após duas iterações, e do (b) elemento pertencente à interface, após duas iterações.

O uso apenas da técnica da bissecção generalizada também é abordada em trabalhos como Pelosi [15], cuja pesquisa relaciona-se com a geração eficiente de elementos finitos para propagação de ondas, e no programa HMesh de autoria do Professor Dr. Hugo E. H. Figueroa. A seguir, as Figuras 3.3 (a) e (b) foram geradas no programa HMesh, baseando-se na aplicação dessa técnica.

Outras variações dessa técnica também foram desenvolvidas, como o caminho de propagação da maior aresta (LEPP), introduzida a partir de 1989 por Rivara [16, 17, 18] e também, em 2000, denominada visita por adjacência, encontrada na tese de doutorado de Baude [2].

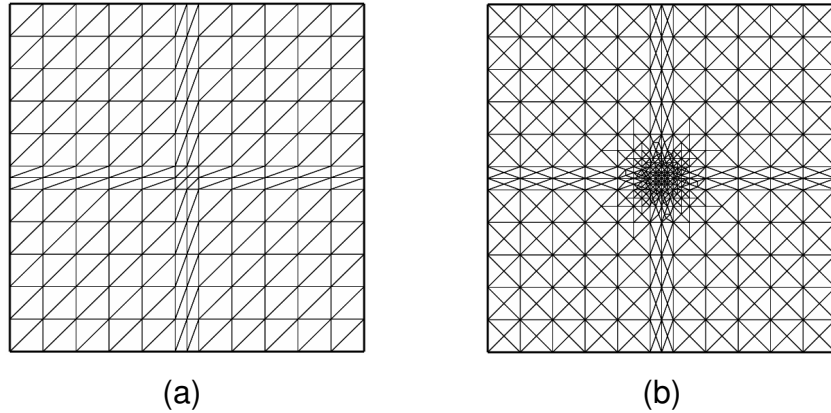


Fig. 3.3: Uso da técnica da bissecção generalizada em uma (a) malha inicial, e o seu (b) refinamento denso localizado no centro de um guia retangular.

Ambas as técnicas baricêntrica e da bissecção, geram malhas com conformidade. As principais vantagens na escolha da técnica baricêntrica são: a melhoria do aspecto da malha, quando em devido uso com a técnica da bissecção, nas interfaces, e a técnica de Delaunay; rapidez na geração da malha, pois uma iteração gera três novos elementos, enquanto na bissecção apenas dois; embora, a técnica da bissecção ofereça malhas de bons aspectos, a sua qualidade está vinculada à qualidade da malha inicial.

Ainda existem outras técnicas de subdivisão de elementos. Dentre elas, a subdivisão em quatro elementos através da bissecção das três arestas do triângulo [16], como mostrada na Figura 3.4 (a); e na Figura 3.4 (b), a sua variação por meio dos pontos médios dos três lados do triângulo [5]. Entretanto, ambas as técnicas, o procedimento de correção dos elementos não conformes demanda tempo computacional e a qualidade final da malha também está diretamente associada à malha inicial [2].

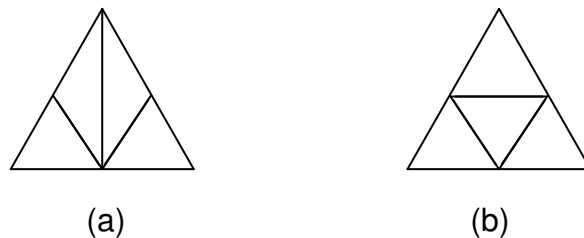


Fig. 3.4: Subdivisão de um elemento em quatro através das técnicas da (a) bissecção dos três lados, e dos (b) pontos médios das três arestas.

Conforme observado, existe uma ampla variedade de técnicas, que podem ser utilizadas na subdivisão dos elementos. Neste trabalho, utilizaram-se as técnicas baricêntrica, e a da bissecção generalizada em condições específicas. A junção dessas duas técnicas de subdivisão em conjuntura com a técnica de Delaunay permitirá maximizar os ângulos mínimos internos dos elementos gerados. Essa triangulação de Delaunay será abordada na próxima seção.

3.2 Triangulação de Delaunay

Nesta seção descreve-se o conceito de arestas de Delaunay e a triangulação de Delaunay. Considera-se um conjunto finito de pontos ou vértices, $V = \{v_0, v_1, v_2, \dots, v_n\}$, $v_i \in \mathbb{R}^2, 0 \leq i \leq n$. De acordo com Shewchuk [19], uma triangulação de V é um conjunto de triângulos $T = \{t_0, t_1, t_2, \dots, t_m\}$, tal que:

- Todo ponto $v_i \in V, 0 \leq i \leq n$ é um vértice de um triângulo $t_k \in T, 0 \leq k \leq m$;
- Para todo $k \neq l, 0 \leq k, l \leq m$, $\text{interior}(t_k) \cap \text{interior}(t_l) = \emptyset$;
- $\bigcup_{k=0}^m t_k = \text{convexo}(V)$,

onde $\text{interior}(t_k)$ denota o interior do k -ésimo triângulo de T e $\text{convexo}(V)$ denota o polígono convexo de V .

A fim de identificar uma aresta de Delaunay, têm-se como processo ilustrativo as Figuras 3.5 (a), (b) e (c). A aresta de Delaunay é encontrada, quando um círculo contém uma aresta e , e nenhum outro vértice está contido neste círculo. No entanto, sabe-se que há várias formas de se dispor um circuncírculo de uma aresta. Portanto, uma aresta é Delaunay, quando existe uma possibilidade de seu circuncírculo, que não contenha um vértice interno a ele. O círculo, mostrado na Figura 3.5 (a), contém a aresta e e um vértice interno a ele. Na Figura 3.5 (b), quando se aumenta o raio do mesmo, ainda é possível encontrar um vértice interno. O raio é aumentado, até que finalmente, na Figura 3.5 (c), nenhum outro

vértice esteja contido. Como a aresta e pertence a uma interface externa, não há um elemento adjacente. Portanto, nessa disposição, é fácil verificar que a aresta e é Delaunay.

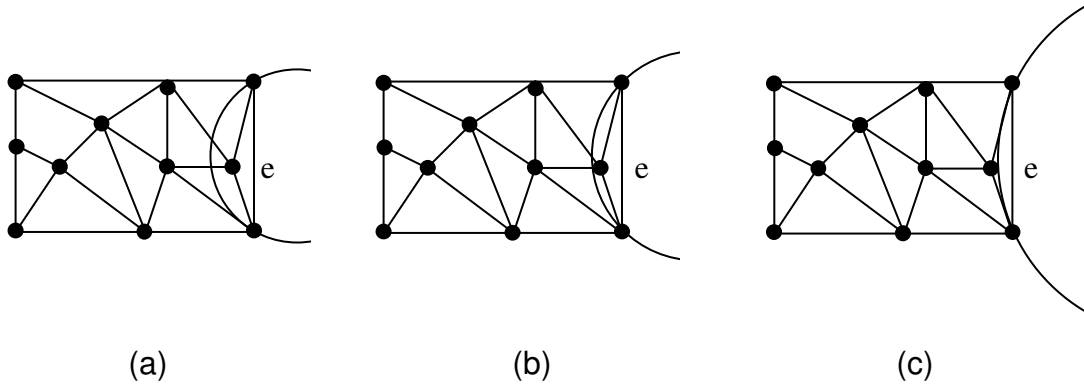


Fig. 3.5: (a) Disposição inicial de um circuncírculo da aresta e , com um vértice interno a ele, e (b) aumentando-se o raio do mesmo, até que se encontre um (b) circuncírculo vazio.

Segundo Shewchuk [12], considerando que T é a triangulação, se todos os triângulos de T são Delaunay, então todas as arestas de T são Delaunay, e vice-versa. A interpretação dada para esta propriedade, é que se todos os triângulos de T são Delaunay, então o circuncírculo de qualquer triângulo é vazio, como mostrado na Figura 3.6 (a). E como todas as arestas de T pertencem a um triângulo de T , todas as arestas estão contidas em um círculo vazio. Portanto, essas arestas são Delaunay.

Para compreender melhor esse processo, considera-se que todas as arestas são Delaunay, e em contraponto, alguns triângulos t de T não são Delaunay. Na Figura 3.6 (b), alguns vértices v de T posicionam-se dentro do circuncírculo de t , no entanto fora do triângulo t . Considera-se e a aresta de t que separa v do interior de t e w o vértice de t oposto a e . Tem-se que o círculo tracejado que contém a aresta e não poderia conter v e nem w . No entanto, como t não é Delaunay, supõe-se que exista um ponto no interior do circuncírculo de t . A Figura 3.6 (b) mostra v no interior deste circuncírculo. Dessa forma, a aresta e não é Delaunay, como se afirmava no início. Sabe-se que na

triangulação T , t não pode conter nenhum outro ponto, desconsiderando os próprios vértices de t , dentro do circuncírculo.

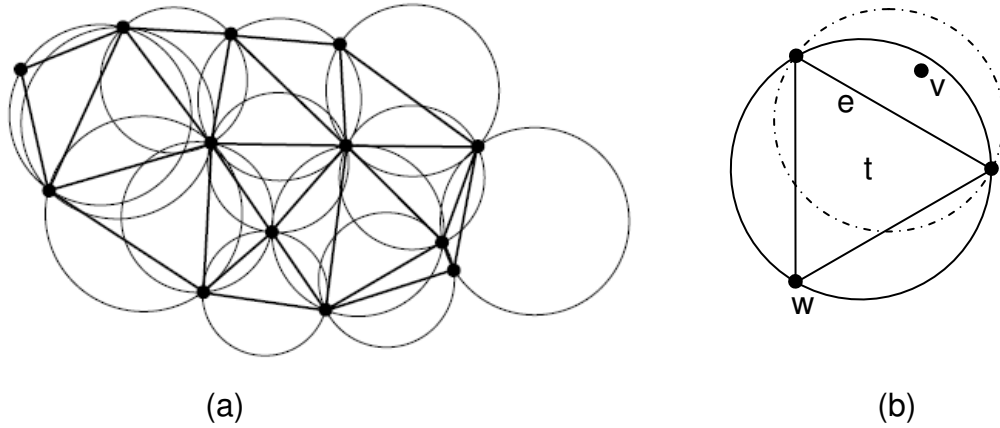


Fig. 3.6: Exemplo de uma (a) triangulação de Delaunay, e quando um (b) triângulo não é Delaunay.

A descrição, a seguir, está relacionada ao processo de troca de arestas ou também conhecido como *edge-flip* [20]. Sempre que o algoritmo de Delaunay identifica uma aresta, que não seja Delaunay, então ela deve sofrer uma troca. O exemplo da Figura 3.7 (a) mostra dois vértices internos à aresta e , e, portanto, a necessidade da troca. Assim, deve-se manter o quadrilátero convexo formado pelos dois triângulos adjacentes, e, em seguida, a aresta e deve sofrer uma exclusão. Por fim, cria-se uma nova aresta e' que a cruza, conforme mostrada na Figura 3.7 (b). Este processo mostra que todas as arestas, produzidas na Figura 3.7 (b), da triangulação final são Delaunay.

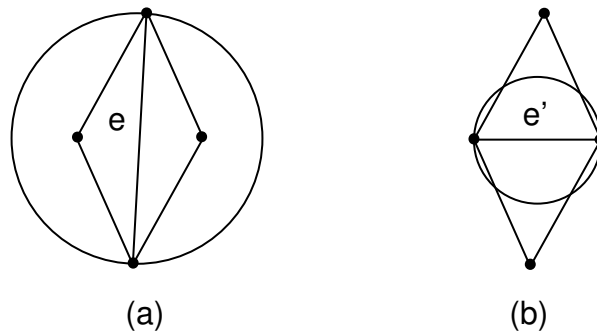


Fig. 3.7: Troca de uma (a) aresta não Delaunay e , pela (b) aresta de Delaunay e' .

No exemplo da Figura 3.8 (a), v e w são os vértices opostos da aresta e . Os vértices v , w , e os pontos extremos de e definem um quadrilátero. C é a circunferência que passa por v e pelos extremos de e . Por meio dessa configuração, temos a possibilidade de w estar estritamente dentro, sobre, ou fora da circunferência C . Caso w esteja sobre ou fora de C , como ilustrado na Figura 3.8 (a), o circuncírculo vazio de C mostra que e é uma aresta de Delaunay. Caso w esteja dentro de C , w deve estar contido dentro do semicírculo, região destacada pela Figura 3.8 (b). No quadrilátero convexo, e deve sofrer o processo de troca de arestas. Com isso, o círculo tracejado que passa por v e w (pontos extremos da nova aresta e'), e tangente a C em v , não contém os extremos da antiga aresta e . Portanto, a aresta e' é Delaunay. Em suma, ou a aresta e de uma triangulação de V é Delaunay, ou deve passar pelo processo de troca de arestas.

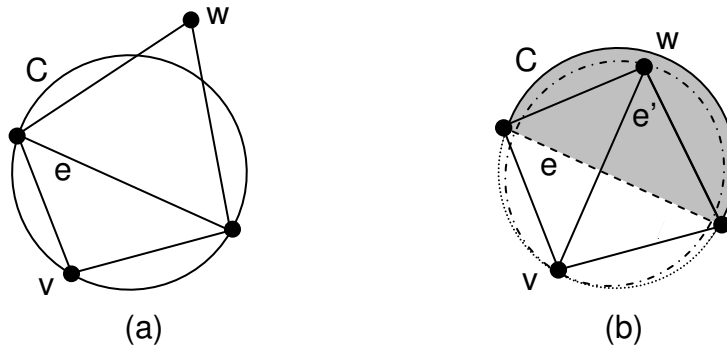


Fig. 3.8: Exemplo de uma triangulação de (a) Delaunay, e (b) não Delaunay ocorrendo a troca de arestas.

A técnica de Delaunay 2-D maximiza o ângulo mínimo interno da triangulação [12, 21]. A Figura 3.9 (a) mostra dois triângulos (p_i, p_j, p_l) e (p_i, p_k, p_j) , e os ângulos internos $\alpha_1 \dots \alpha_6$. Após a troca de arestas, mostrada na Figura 3.9 (b), tem-se os dois novos triângulos (p_i, p_k, p_l) e (p_l, p_k, p_j) , e os novos ângulos internos $\alpha'_1 \dots \alpha'_6$. Ambas as figuras mostram que $\min(\alpha_i) \leq \min(\alpha'_j)$, para $1 \leq i, j \leq 6$.

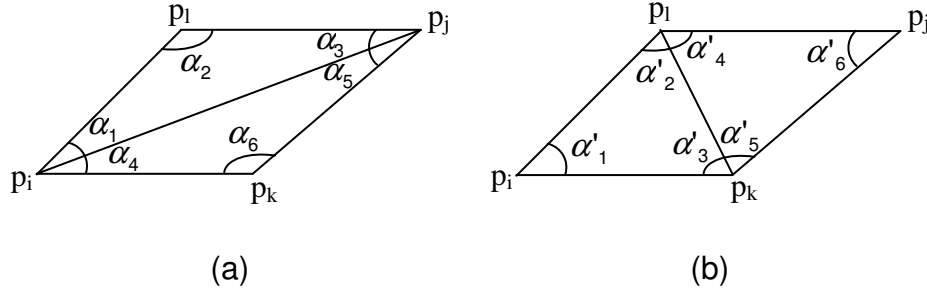


Fig. 3.9: (a) Antes da troca de arestas, e (b) após, com a maximização dos ângulos mínimos internos.

No entanto, a técnica de Delaunay possui dois problemas que devem se levados em consideração. Primeiro, existe uma exceção em domínios bidimensionais, quando a combinação dos triângulos não forma um polígono convexo, o que pode provocar uma degradação na interface do objeto. Considera-se uma malha inicial, mostrada na Figura 3.10 (a), a partir de uma geometria de entrada côncava.

A função *inCircle()*, que será vista na Seção 3.4, é responsável pela checagem de vértices internos ou externos ao circuncírculo de um elemento. Esse processo pode ser acompanhado na Figura 3.10 (b), onde se verifica que, existe um vértice interno ao circuncírculo de t_1 . A primeira impressão é que, nessa configuração, deveria ocorrer uma troca de aresta, como visto na Figura 3.10 (c). No entanto, essa troca não deveria ocorrer, levando-se em consideração apenas uma simples checagem de vértice contido no circuncírculo de t_1 . É necessário verificar também se este vértice pertence a um elemento adjacente de t_1 . Se isto não for analisado, tem-se o seguinte resultado ilustrado na Figura 3.11 (c). Esta figura mostra a interface excluída, representada pela linha tracejada, e a estrutura final define uma geometria diferente da entrada, não côncava. Portanto, ressalta-se que é necessário utilizar corretamente a troca de arestas de modo que não exista degradação da interface inicial. Existem diversos métodos que lidam com a questão da integridade das bordas e que pode ser obtido em trabalhos como [21, 22].

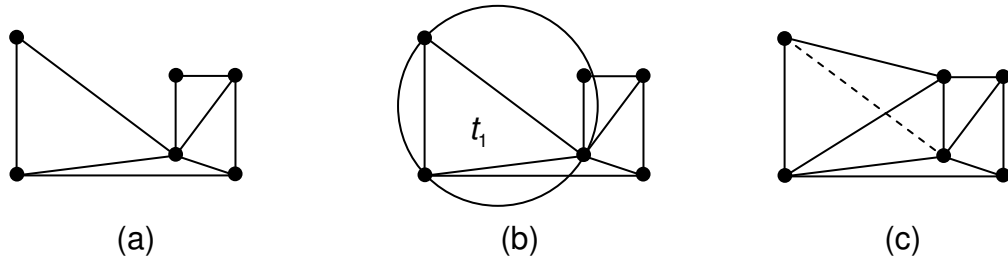


Fig. 3.10: (a) Geração de uma malha inicial côncava; a (b) checagem de um vértice interno a t_1 ; e a (c) degradação da interface.

O segundo problema, é que a técnica de Delaunay não possui um tratamento nos triângulos da interface. A Figura 3.11 (a) mostra um triângulo t_2 degenerado. Como não existe qualquer vértice interno ao circuncírculo de t_2 , já que este se encontra na borda da estrutura, não haverá troca de arestas. Portanto, o triângulo t_2 permanecerá degenerado.

Existem dois modos de contornar esse problema: a adição de novos vértices ao longo da interface, mostrada na Figura 3.11 (b), e realizar uma nova triangulação a partir desses vértices; ou utilizar a técnica da bissecção generalizada, que reduz os elementos em partes menores, e a partir disso realizar a troca de arestas. Neste trabalho procurou-se solucionar através do segundo modo, e essa implementação será melhor compreendida no Capítulo 4.

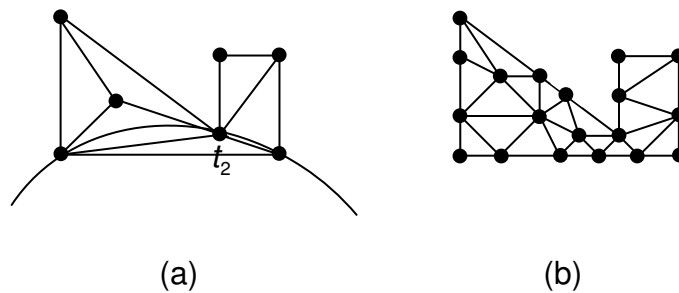


Fig. 3.11: Recuperação do elemento (a) t_2 de baixa qualidade, após a (b) adição de vértices na sua interface.

3.3 Algoritmos de Delaunay

Existem diversos algoritmos para se construir uma triangulação de Delaunay, dentre eles *edge-flip* (troca de arestas), Bowyer/Watson, *Lawson's incremental insertion* (inserção incremental de Lawson) [10], *divide-and-conquer* (dividir e conquistar) [23], e *sweepline* (troca de linhas) [23]. Os algoritmos de Bowyer/Watson e a inserção incremental de Lawson serão descritos brevemente nesta seção. Os algoritmos dividir e conquistar e a troca de linhas não serão abordados, e podem ser obtidos detalhadamente no trabalho de Shewchuk [12]. A técnica mais simplificada é a troca de arestas, descrita por Gavrilova [23] e associada com a Seção 3.2, que envolve uma transformação na triangulação, excluindo-se a aresta interna de um quadrilátero convexo. Um exemplo de processo do algoritmo de Bowyer-Watson [21] encontra-se ilustrado nas Figuras 3.12 (a) – (c).

Na Figura 3.12 (a), um novo ponto p' é inserido, de forma arbitrária, na malha inicial. Na Figura 3.12 (b), os triângulos, cujos circuncírculos contenham este ponto, não são Delaunay. Dessa forma, esses triângulos são excluídos formando uma cavidade, denominada polígono de inserção. Todos os outros triângulos que são Delaunay permanecem inalterados. Cada ponto do polígono de inserção é conectado ao novo vértice por meio de uma nova aresta, mostrada na Figura 3.12 (c). Assim, o conjunto dessas novas arestas é Delaunay.

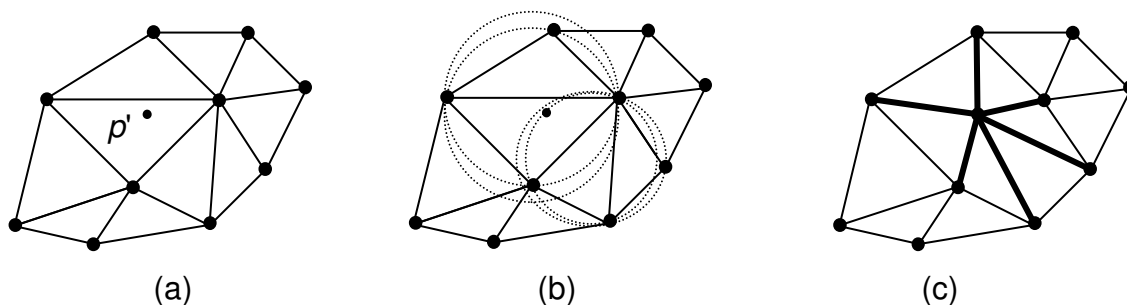


Fig. 3.12: Algoritmo de Bowyer/Watson, na (a) inserção de um novo ponto p' ; na verificação dos (b) triângulos, cujo circuncírculo contenham p' ; e onde as (c) arestas criadas se interceptam em p' .

O algoritmo de Bowyer/Watson, como pôde ser observado, produz uma triangulação de Delaunay sem a técnica da troca de arestas. Diferentemente do Bowyer/Watson, o algoritmo de inserção incremental de Lawson [12] insere um vértice de cada vez. As etapas, na inserção de um vértice, podem ser acompanhadas nas Figuras 3.13 (a) – (e).

A Figura 3.13 (a) mostra a inserção de um novo ponto p' , seguida pela Figura 3.13 (b), que mostra a conexão deste vértice com os nós do triângulo a que este pertence. A partir disso, a técnica de Delaunay é efetuada nas Figuras 3.13 (c) – (e). Como se pode observar, tanto a técnica de Bowyer/Watson como a de Lawson, cujos resultados são vistos, respectivamente, em 3.12 (c) e 3.13 (e), produzem resultados finais semelhantes.

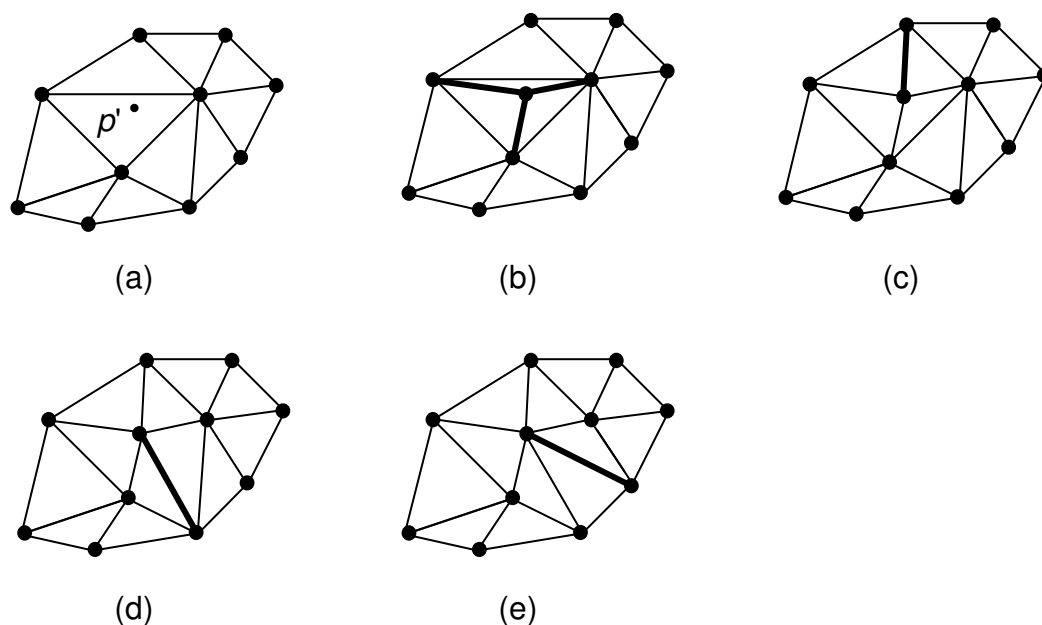


Fig. 3.13: Algoritmo de inserção incremental de Lawson, na (a) inserção de um novo ponto p' ; na (b) formação de novos elementos; e na (c) - (e) troca de arestas.

Neste trabalho, o algoritmo implementado é semelhante ao algoritmo de Lawson, no que se refere ao processo de checagem de vizinhos e troca de arestas quando necessário. Em relação à inserção de um novo nó de cada vez na topologia, em Lawson a localização é arbitrária, enquanto neste trabalho, o novo

vértice estará sempre posicionado no centróide do elemento, utilizando-se a técnica baricêntrica.

Por fim, na prática, a técnica de Delaunay está ligada aos predicados geométricos, onde dada uma triangulação, os predicados informam se o circuncírculo de um triângulo possui algum vértice contido nele. Esses predicados são apresentados na próxima seção.

3.4 Predicados Geométricos

Os predicados geométricos consistem de propriedades de orientação com propósito utilizado em algoritmos geométricos. Muitos são utilizados na implementação da triangulação de Delaunay, e podem ser obtidos em trabalhos como de Shewchuk [12] e de Gavrilova [23], que tratam especificamente sobre a geração de malhas utilizando a técnica de Delaunay. Segundo Schewchuk [12], sejam a , b , c , d quatro pontos num plano, define-se uma função denominada $orient2D(a, b, c)$ que retorna as seguintes informações: um valor positivo se os pontos a , b , c estão ordenados em sentido anti-horário; um valor negativo se os pontos estão no sentido horário; e zero se os pontos são colineares. A interpretação mais comum para a função $orient2D$ é: retornar um valor positivo, se a permanece à esquerda do segmento \overline{bc} ; e negativo, se a permanece à direita do mesmo. Esse cálculo é obtido em (3.1).

$$Orient2D(a, b, c) = \begin{vmatrix} a_x & a_y & 1 \\ b_x & b_y & 1 \\ c_x & c_y & 1 \end{vmatrix} = \begin{vmatrix} a_x - c_x & a_y - c_y \\ b_x - c_x & b_y - c_y \end{vmatrix} \quad (3.1)$$

Define-se também um procedimento denominado $inCircle(a, b, c, d)$, que retorna um valor positivo se d permanece dentro do circuncírculo definido pelos vértices a , b , c . Esse circuncírculo deve seguir sempre uma orientação, portanto, os pontos a , b , c , definidos nesse projeto, devem ter sentido anti-horário. A função $inCircle(a, b, c, d)$ retorna zero, se os quatro pontos permanecem sobre o circuncírculo. E negativo, se d está fora do circuncírculo.

$$inCircle(a, b, c, d) = \begin{vmatrix} a_x & a_y & a_x^2 + a_y^2 & 1 \\ b_x & b_y & b_x^2 + b_y^2 & 1 \\ c_x & c_y & c_x^2 + c_y^2 & 1 \\ d_x & d_y & d_x^2 + d_y^2 & 1 \end{vmatrix} = \begin{vmatrix} a_x - d_x & a_y - d_y & (a_x^2 + a_y^2) - (d_x^2 + d_y^2) \\ b_x - d_x & b_y - d_y & (b_x^2 + b_y^2) - (d_x^2 + d_y^2) \\ c_x - d_x & c_y - d_y & (c_x^2 + c_y^2) - (d_x^2 + d_y^2) \end{vmatrix} \quad (3.2)$$

Quando os pontos estiverem no sentido horário, a função será reversa pela troca de sinais, como se a região externa à circunferência fosse a própria região interna e vice-versa.

A Tabela 3.1 mostra a integração de (3.2) com o algoritmo de troca de arestas. Note, nas linhas 3 – 5, que os pontos a , b , c possuem as coordenadas dos três nós de um elemento i . A linha 6 faz uma busca dos vizinhos do elemento i ; e a linha 7 indica d como o ponto útil de cada vizinho; para posterior troca de arestas em relação ao elemento i , mostrada na linha 9.

Algoritmo: *inCircle test*

1. para $i = 1 \dots Nelem_m$
2. pesquisar os pontos p_1^i , p_2^i e p_3^i
3. $a = (p_1^i x, p_1^i y)$
4. $b = (p_2^i x, p_2^i y)$
5. $c = (p_3^i x, p_3^i y)$
6. pesquisar os vizinhos do $elemento_i = \{viz_1^i, viz_2^i, viz_3^i\}$
7. $d = (p_viz_j^i x, p_viz_j^i y)$
8. se $inCircle = (a, b, c, d) > 0$
9. *edge-flip*
10. fim se
11. fim para

Tab. 3.1: Algoritmo de teste de nó dentro do circuncírculo.

Ao longo deste trabalho, observou-se o estudo a respeito de algumas técnicas, como a função densidade, a subdivisão de um elemento e a triangulação de Delaunay. Entretanto, ainda se torna necessário, um estudo na avaliação da qualidade dos elementos gerados no programa. Assim, pretende-se partir para o estudo a respeito dos aspectos qualitativos de uma malha. Os critérios utilizados são os cálculos dos fatores de qualidade retiradas da literatura. Esse estudo é

relevante para se obter informações condizentes com o objetivo deste estudo, ou seja, na geração de elementos próximos das condições regulares.

3.5 Aspectos Qualitativos de uma Malha

A qualidade de um elemento está diretamente associada com o grau de seus ângulos internos e a medida de suas arestas. Quando todas as arestas de um elemento possuem comprimentos iguais, o elemento é dito regular. Como será visto nesta seção, o grau de achatamento do elemento está associado aos ângulos e comprimento das arestas, os quais definem a sua qualidade. Os cálculos dos fatores de qualidade apresentados são: a razão entre o raio circunscrito e a menor aresta; a razão entre os raios inscrito e circunscrito; e outros critérios de qualidade.

3.5.1 Razão ente Raio Circunscrito e Menor Aresta

Existe uma variedade de medidas que podem ser obtidas em Shewchuk [24]. Para se compreender a primeira relação ente a qualidade e a geometria do elemento, tem-se como exemplo de medida, a razão entre o raio circunscrito R e o comprimento da menor aresta e . Esta relação, dado um triângulo na k -ésima dimensão, é mostrada em (3.3) [3].

$$\frac{R}{e} = \sqrt{\frac{k}{2(k+1)}} \quad (3.3)$$

Dado um triângulo de lados d_1 , d_2 e d_3 , o cálculo do fator de qualidade é obtido através da seguinte expressão:

$$Q_{\frac{R}{e}} = \sqrt{3} \cdot \frac{\prod_{i=1}^3 d_i}{e \cdot (4 \cdot \sqrt{s(s-d_1)(s-d_2)(s-d_3)})}, \quad (3.4)$$

onde $\sqrt{3}$ é o valor de normalização obtido através de (3.3) para $k=2$ (bidimensional), e s é o cálculo do semi-perímetro do triângulo.

Em 2-D também se pode obter a mesma relação em função do ângulo mínimo. A Figura 3.14 (a) mostra um dado $\Delta_{p_i p_j p_k}$ de circuncentro O e o raio do circuncírculo R . Tem-se que a aresta $\overline{p_i p_j}$ é dada por $e = 2R \sin \alpha_{\min}$, e seu ângulo oposto é $\alpha_{\min} = \angle p_i p_k p_j$. Dessa forma, utilizando-se a lei dos senos, a razão entre o raio circunscrito e menor aresta é visto em (3.5).

$$\frac{R}{e} = \frac{1}{2 \sin(\alpha_{\min})} \quad (3.5)$$

A partir de (3.5), observa-se que quanto maior o ângulo α_{\min} , menor é a razão. Através do gráfico, ilustrado na Figura 3.15 (b), pode-se observar essa maximização de α_{\min} em função da minimização de $\frac{R}{e}$.

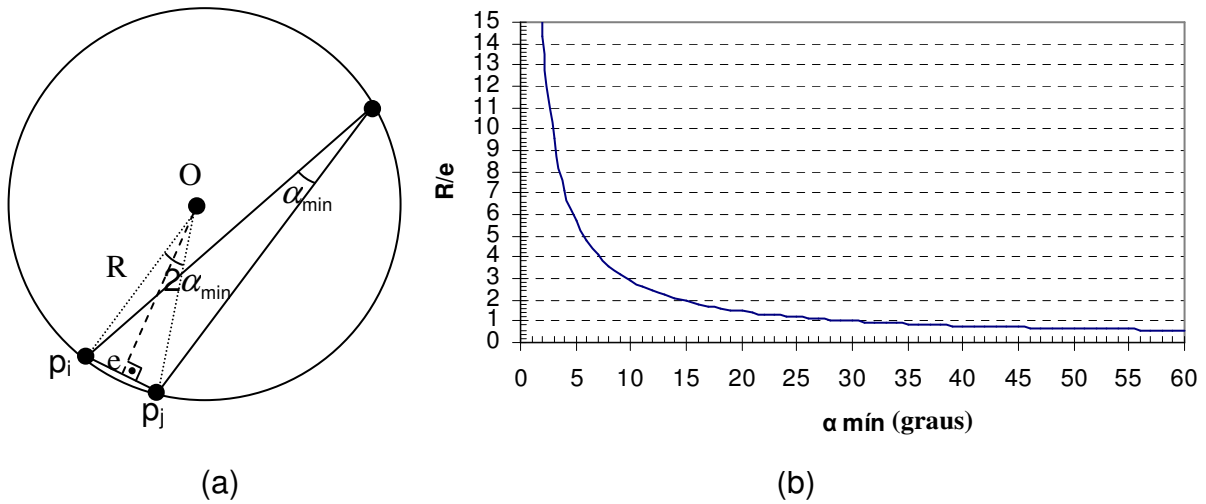


Fig. 3.14: (a) Relação da razão entre o raio circunscrito e a menor aresta através de α_{\min} , e o (b) gráfico de variação de $\frac{R}{e}$ em função de α_{\min} .

O menor valor da razão de (3.5) é dado por $\alpha_{\min} = \frac{\pi}{3} \text{ rad} = 60^\circ$, cujo valor substituído em α_{\min} , obtém-se (3.6).

$$\frac{R}{e} = \frac{1}{2 \sin\left(\frac{\pi}{3}\right)} = \frac{1}{\sqrt{3}} \approx 0,577; \quad (3.6)$$

onde o valor obtido é coerente com o resultado visto em (3.3). Esse valor também pode ser visto pelo gráfico da Figura 3.14 (b). Para $\alpha_{\min} = \frac{\pi}{3} \text{ rad}$, $\frac{R}{e} \approx 0,577$. Esse gráfico mostra que para $\alpha_{\min} = \frac{\pi}{3} \text{ rad}$, o triângulo é dito regular, e à medida que o ângulo diminui, o elemento se aproxima da degeneração.

Se considerar B uma constante da razão $\frac{R}{e}$, definir-se-á a existência de ângulos maiores que α_{\min} e menores que α_{\max} . Esses ângulos são obtidos através das equações mostradas em (3.7).

$$\alpha_{\min} = \arcsin\left(\frac{1}{2B}\right), \alpha_{\max} = 180 - 2\arcsin\left(\frac{1}{2B}\right) \quad (3.7)$$

A Tabela 3.2 mostra um exemplo da constante B e a relação de ângulos mínimos e máximos. Quanto maior a constante B , menor o ângulo mínimo e maior o ângulo máximo. Entretanto, para valores distantes de 0,577, a consideração por elementos degenerados aumenta. Por exemplo, para $B=1$, os elementos terão ângulos mínimo $\alpha_{\min} \approx 30^\circ$ e máximo $\alpha_{\max} \approx 120^\circ$.

$B \left(= \frac{R}{e} \right)$	α_{\min}	α_{\max}
0,577	60°	60°
1	$\approx 30^\circ$	$\approx 120^\circ$
$\sqrt{2}$	$\approx 20,7^\circ$	$\approx 138,6^\circ$
$\sqrt{3}$	$\approx 16,77^\circ$	$\approx 146,46^\circ$

Tab. 3.2: Tabela de cálculo de B , α_{\min} e α_{\max} .

Esse limite pode ser definido em algoritmos de Delaunay, como Bowyer/Watson, na inserção de um vértice no elemento. Alguns algoritmos consideram $B = 1$, como em [12], como limite para avaliar a inserção de um nó. O limite B está associado com a inserção de vértice, visto na Figura 3.12, para avaliar a qualidade dos elementos gerados.

3.5.2 Razão entre Raios Inscrito e Circunscrito

O fator de qualidade normalizado de um elemento regular possui o valor um, e de um elemento degenerado, o valor zero. Uma medida de qualidade pode ser obtida pela razão entre os raios dos círculos inscrito e circunscrito ao elemento [25], [26], [27]. Dado um triângulo de lados d_1 , d_2 , d_3 , para um triângulo regular na k -ésima dimensão, a razão entre o raio inscrito r e o circunscrito R é dada pela relação (3.8) [3]:

$$\frac{r}{R} = \frac{1}{k} \quad (3.8)$$

A partir de (3.8), obtém-se o fator de qualidade $Q_{\frac{r}{R}}$, normalizado para k dimensão, que pode ser visto em (3.9). Assim, uma menor razão, de $Q_{\frac{r}{R}}$, indica maior o raio circunscrito e menor o raio inscrito. Portanto, maior será o achatamento do elemento para $k=2$.

$$Q_{\frac{r}{R}} = k \frac{r}{R} \quad (3.9)$$

Para um simplexo 2-D, a equação (3.9) também pode ser escrita como:

$$Q_{\frac{r}{R}} = 8 \frac{(s-d_1)(s-d_2)(s-d_3)}{\prod_{i=1}^3 d_i}, \text{ onde } s = \frac{\sum_{i=1}^3 d_i}{2} \quad (3.10)$$

No trabalho de Ribeiro Filho [5], os fatores de qualidade mostrados em (3.9) e (3.10) são tratados como critérios distintos. No entanto, através das fórmulas dos raios inscrito e circunscrito, obtidos em [28], e substituindo-as na equação (3.9) obtém-se (3.10). Portanto, os dois fatores produzem o mesmo resultado normalizado para $k=2$.

3.5.3 Outros Critérios de Qualidade

Em (3.11), dado um triângulo de lados d_1, d_2, d_3 , o fator de qualidade α é calculado através da área A do elemento, dividido pela soma dos quadrados dos lados. Este fator produz resultados semelhantes a da razão dos raios inscrito e circunscrito [24], embora, menos conservador. Como será observado ao longo deste trabalho, optou-se pela escolha deste fator de qualidade como critério de avaliação dos elementos.

$$\alpha = 4\sqrt{3} \frac{A}{\sum_{i=1}^3 d_i^2} \quad (3.11)$$

Ademais, um novo critério introduzido por [5], é dado pelo fator de qualidade β , visto em (3.12). Esta medida realiza uma comparação entre a área do triângulo A_T , de lados d_1, d_2, d_3 ; e a área de um triângulo regular A_L , cujos lados têm comprimento L , onde L é a média geométrica dos lados d_1, d_2 e d_3 .

$$\beta = \frac{A_T}{A_L}, \text{ onde } L = \left(\prod_{i=1}^3 d_i \right)^{\frac{1}{3}} \quad (3.12)$$

Segundo Ribeiro Filho [5], os fatores de qualidade α , β e $\frac{r}{R}$, nos testes gerados, demonstraram que $\beta > \alpha$ e $\alpha > \frac{r}{R}$, somente quando o triângulo é irregular. Como medida de teste, tem-se a Figura 3.15, que ilustra um triângulo p_i, p_j, p_k regular de lado dois com a variação do ponto p_k ao longo do eixo y. Dessa forma, as arestas $\overline{p_k p_j}$ e $\overline{p_k p_i}$, representadas pelas linhas tracejadas, sofrerão deslocamento possibilitando a análise de elementos irregulares.

A Tabela 3.3 mostra essa variação de $p_k(y)$ de -0,5 até 7,0; com razão 0,5. Para cada posição de $p_k(y)$, calculam-se os respectivos fatores de qualidade normalizados para α , $\frac{r}{R}$ e β . Dessa maneira, tomando como referência o

triângulo p_i , p_j , p_k regular, para $p_k(y)=1,0$, os três fatores de qualidade possuem o valor 1,0.

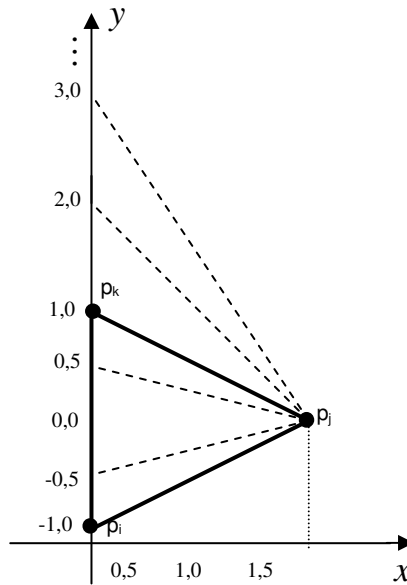


Fig. 3.15: Triângulo p_i , p_j , p_k , com o deslocamento do vértice $p_k(y)$.

$p_k(y)$	α	r/R	β
-0,5	0,4	0,38675	0,675106
0,0	0,75	0,732051	0,87358
0,5	0,947368	0,941451	0,973672
1,0	1,0	1,0	1,0
1,5	0,967742	0,963961	0,983868
2,0	0,9	0,889822	0,949914
2,5	0,823529	0,808389	0,911281
3,0	0,75	0,732051	0,87358
3,5	0,683544	0,664479	0,838798
4,0	0,625	0,60591	0,807388
4,5	0,573913	0,555428	0,779214
5,0	0,529412	0,511858	0,753947
5,5	0,490566	0,474087	0,731224
6,0	0,456522	0,441153	0,710702
6,5	0,42654	0,412259	0,692082
7,0	0,4	0,38675	0,675106

7,5	0,376384	0,364097	0,659559
8,0	0,355263	0,343864	0,645257
8,5	0,336283	0,325698	0,632047
9,0	0,319149	0,309307	0,619798
9,5	0,303614	0,294451	0,608401
10,0	0,289474	0,280928	0,597762
10,5	0,276553	0,268571	0,5878
11,0	0,264706	0,257237	0,578446
11,5	0,253807	0,246807	0,569641
12,0	0,24375	0,237179	0,561331
12,5	0,234443	0,228265	0,553471
13,0	0,225806	0,219989	0,546022
13,5	0,217772	0,212285	0,538948
14,0	0,21028	0,205098	0,532218
14,5	0,203279	0,198377	0,525804
15,0	0,196721	0,192079	0,519683

Tab. 3.3: Tabela de cálculo do fator de qualidade normalizado entre α , $\frac{r}{R}$ e β .

O gráfico, gerado pela Figura 3.16, mostra a variação da curva dos três fatores a partir da Tabela 3.3. Nota-se que α e $\frac{r}{R}$ sempre mantêm uma relação proporcional de variação do vértice p_k em função do fator de qualidade. Entretanto, o fator β apresenta uma relação de distanciamento da curva, que está acima da qualidade definida por α e $\frac{r}{R}$. Sabe-se que o fator de qualidade normalizado varia de 0 a 1, respectivamente, de pior ao melhor caso. Para elementos classificados como degenerados pelo critério α e $\frac{r}{R}$, diferentemente em β , os elementos são avaliados como de qualidade aceitável, por exemplo, para $\alpha \cong \frac{r}{R} \cong 0,2$, tem-se que $\beta \cong 0,5$. Os resultados mostram que o fator de qualidade β pode trazer resultados incoerentes na avaliação qualitativa de uma malha.

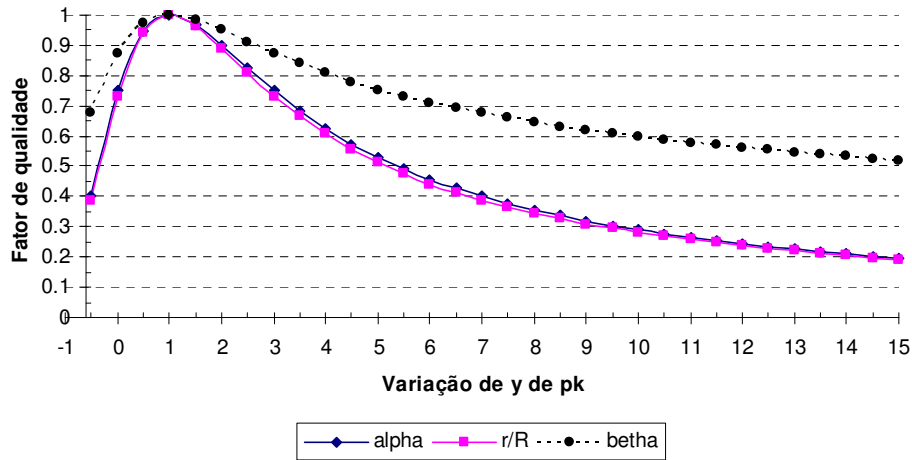


Fig. 3.16: Gráfico de variação de $p_k(y)$ em função de α , $\frac{r}{R}$ e β .

Segundo Lee [29], existe uma relação que calcula a qualidade geral de uma malha, dada por $\bar{\alpha}$. Esse fator normalizado, obtido em (3.13), realiza uma média geométrica da qualidade individual α_i , para $i = 1, \dots, n$, onde n é o número total de elementos de uma malha.

$$\bar{\alpha} = \left(\prod_{i=1}^n \alpha_i \right)^{\frac{1}{n}} \quad (3.13)$$

Conforme observado, o fator de qualidade serve como critério de avaliação dos elementos em uma determinada malha. Ainda assim, caso exista a necessidade de se aperfeiçoar tal qualidade, deve-se buscar técnicas, como o Delaunay e a suavização de Laplace, que conduzem para uma melhoria do aspecto qualitativo. A partir da próxima seção, será abordada a suavização de Laplace e a sua variação restrita.

3.6 Técnicas de Suavização

A suavização laplaciana é a técnica mais utilizada para melhorar a suavização da malha. Por ser computacionalmente eficaz, ela simplesmente desloca cada nó ao centróide do polígono, que é formado pelos nós adjacentes. O algoritmo de suavização é considerado local, pois em cada iteração, o deslocamento do nó é calculado usando-se apenas os nós adjacentes. É comum usá-lo somente em algumas iterações porque a qualidade dos elementos não se modifica consideravelmente após a malha ter atingido um equilíbrio. Além disso, a sua aplicação nem sempre é adequada, podendo muitas vezes piorar a qualidade do elemento [30].

Considera-se, como exemplo, uma malha ilustrada na Figura 3.17. Observa-se o nó (x, y) no interior do polígono, que é formado pelos vértices (x_i, y_i) , para $1 \leq i \leq N$, onde N é o número total de nós. Cada aresta conecta o nó central (x, y) aos seus nós adjacentes. A suavização laplaciana é vista como uma técnica de otimização local dos nós, e após a aplicação de (3.14), o novo nó corresponderá à nova posição (x', y') .

$$x' = \frac{1}{k} \sum_{i=1}^N x_i, \quad y' = \frac{1}{k} \sum_{i=1}^N y_i \quad (3.14)$$

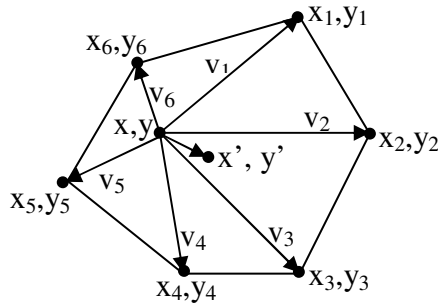


Fig. 3.17: Equilíbrio de um nó local.

O algoritmo da suavização laplaciana pode ser visto na Tabela 3.4. Nele, têm-se como entrada os vértices adjacentes (x_i, y_i) que formam o polígono convexo. Na linha 1 é necessário definir o número de iterações $N_{\text{iteração}}$ desejado na malha. Na linha 2, realiza-se o cálculo da nova posição, e, em seguida, é necessário atualizar a nova coordenada do vértice central do polígono, mostrado na linha 4.

Algoritmo: Suavização laplaciana

Entrada: Vértices (x_i, y_i) adjacentes ao nó (x, y)

1. para $n = 1 \dots N_{\text{iteração}}$

$$2. \quad x' = \frac{1}{k} \sum_{i=1}^N x_i \text{ e } y' = \frac{1}{k} \sum_{i=1}^N y_i$$

3. fim para

4. atualizar (x, y) para (x', y')

Tab. 3.4: Algoritmo de suavização laplaciana.

A principal vantagem desse algoritmo é a eficiência computacional e facilidade de implementação. Entretanto, existem algumas desvantagens relacionadas a esse método: nem sempre desloca o nó central para a posição ótima, e, portanto, nem sempre melhora a qualidade dos elementos; e dependendo da aplicação, podem-se gerar elementos invertidos, por exemplo, tendo um polígono não convexo formado por n elementos, o centróide desse polígono, após a suavização, poderá se deslocar para a região externa à este

polígono; é possível também que quanto maior o número de iterações, maior a possibilidade de diminuir a qualidade dos elementos.

Uma variante da suavização descrita anteriormente, é a suavização laplaciana restrita (*Smart Laplacian Smoothing*) vista em trabalhos como de Freitag [31] e de Borouchaki [32]. Esta técnica consiste em deslocar o ponto geométrico somente se a medida de qualidade local na malha é melhorada. O algoritmo pode ser visto na Tabela 3.5. A diferença desse algoritmo está na seguinte condição, mostrado na linha 6: caso a qualidade final normalizada dos elementos, mediante a nova posição do vértice central, seja maior que a qualidade inicial normalizada dos elementos, então esse vértice deverá sofrer atualização, como pode ser visto na linha 7. Caso contrário, o vértice permanecerá inalterado.

Algoritmo: Suavização laplaciana restrita
 Entrada: Vértices (x_i, y_i) adjacentes ao nó (x, y)

1. calcular $Q_{inicial}$ dos elementos comuns ao nó (x, y)
2. para $n = 1 \dots N_{iteração}$
3. $x' = \frac{1}{k} \sum_{i=1}^k x_i$ e $y' = \frac{1}{k} \sum_{i=1}^k y_i$
4. fim para
5. calcular Q_{final} dos elementos comuns ao nó (x', y')
6. se $Q_{final} > Q_{inicial}$ então
7. atualizar (x, y) para (x', y')
8. fim se

Tab. 3.5: Algoritmo de suavização laplaciana restrita.

Finalmente, existem outras técnicas interessantes, as quais não serão abordadas, como a suavização baseada em ângulos, vista no trabalho de Zhou [30], que realiza o cálculo da nova posição do nó baseada na checagem dos ângulos formados pelos cantos do polígono convexo; e a relaxação, abordada em trabalhos como de Frey [33] e [34], que realiza um cálculo da quantidade de arestas conectadas a um determinado nó.

Capítulo 4

Implementação Computacional

Este capítulo descreve a implementação da função de densidade, apresentada na Seção 2.3, e o modelo sistemático de decomposição de uma malha inicial através da combinação das técnicas baricêntrica, de Delaunay, e da suavização laplaciana. Apresentam-se, também, soluções para a integridade geométrica das interfaces retangular, circular e curva arbitrária, utilizando-se os algoritmos de integridade de bordas. Estes são responsáveis por evitar a degradação da interface, também conhecido como buracos topológicos, causada pelas sucessivas subdivisões.

4.1 Algoritmo de Subdivisão dos Elementos

A abordagem prática da função de densidade, mostrada na Seção 2.3, pode ser compreendida por meio do algoritmo apresentado na Tabela 4.1. Nessa tabela, é mostrada uma conversão, para o formato algorítmico, partindo-se do fluxograma observado na Figura 4.1 do trabalho realizado por Baude [2].

Na Tabela 2.2 tem-se como variável de entrada $Nelem_{estimado}$, que é a quantidade de elementos estimados pelo usuário. A condição de parada do algoritmo pode ser vista na linha 3, onde $Nelem_m$ é o número de elementos produzidos numa iteração m do algoritmo. A quebra de elementos é vista na

linha 5, ou seja, se o peso individual w_i de um elemento i for maior que a densidade por elemento VE_m , então, conforme visto na linha 6, o elemento i deve sofrer subdivisão. A linha 7 mostra que após a criação dos novos elementos, o peso individual w_i deverá ser recalculado novamente. Por fim, a linha 11 indica que a densidade por elemento VE_m também deverá ser calculada novamente.

No algoritmo de subdivisão, é importante ressaltar que o critério de parada deste processo iterativo, concentra-se na variável de entrada $Nelem_{estimado}$, que é um valor definido pelo pesquisador e baseia-se na densidade de refinamento que se deseja para uma determinada estrutura. Tal critério fora utilizado partindo-se da experiência do pesquisador no estudo de propagação de problemas eletromagnéticos. Da mesma maneira que outros programas, como o GiD [35], procuram estabelecer uma estimativa inicial, onde é fornecido o tamanho da aresta do elemento, neste projeto, utilizou-se o número estimado de elementos, que é um critério abordado em Fernandez [13].

Algoritmo: Subdivisão dos elementos

Entrada: $Nelem_{estimado}$

1. cálculo do peso global VE_m
2. cálculo do peso individual w_i
3. enquanto $Nelem_m < Nelem_{estimado}$
4. para $i = 1 \dots Nelem_m$
5. se $w_i > VE_m$
6. refinamento do elemento i
7. cálculo do peso individual w_i
8. fim se
9. fim para
10. cálculo do número de elementos $Nelem_m$
11. cálculo do peso global VE_m
12. fim enquanto

Tab. 4.1: Algoritmo de subdivisão dos elementos.

Também se ressalta que não existe um número mínimo de elementos estimados, uma vez que se a malha inicial possuir uma determinada quantidade

inicial, e se o número estimado for menor que esta quantidade, a malha não sofrerá qualquer refinamento. Mesmo para uma quantidade um pouco maior que o estimado, após o refinamento, a malha não sofrerá nenhuma degeneração. E a quantidade máxima de elementos gerados baseia-se no valor máximo definido pelos vetores de nós e de elementos, no código do gerador de malhas.

Dando seqüência, o motivo, da conversão para a tabela 4.1, é devido ao fluxograma, mostrado na Figura 4.1, ser utilizado apenas como ferramenta no auxílio da compreensão da lógica de programação, ou seja, utilizar objetos gráficos para representar comandos. No entanto, o fluxograma não possibilita que a lógica seja executada em um computador, necessitando com isso, que o pesquisador imagine a execução do programa. Um fluxograma fica distante da real construção de um código [36], [37]. Por essa razão, considerou-se relevante a conversão do fluxograma na forma algorítmica, para facilitar a compreensão prática.

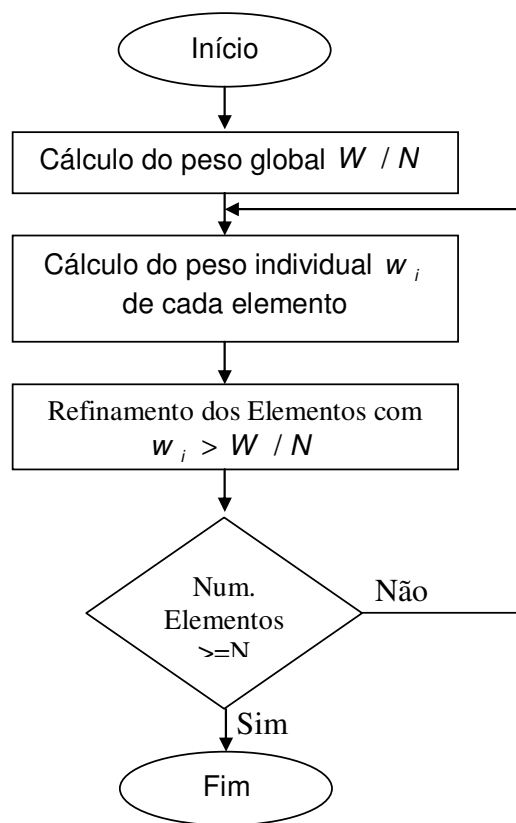


Fig. 4.1: Fluxograma de subdivisão dos elementos.

Salienta-se que a abordagem do algoritmo é parte fundamental, uma vez que se está partindo de uma literatura de programação estruturada, vista em trabalhos como Baude [2], em direção a programação orientada a objeto. Isso permitirá que outros autores possam reproduzir levando as vantagens que esta última oferece [38], [39], [40].

As Figuras a seguir, são exemplos de malhas geradas através da implementação da função de densidade. A Figura 4.2 (a) mostra a curva da equação (2.1), vista na Seção 2.3, gerada no Matlab; e a Figura 4.2 (b) é o mesmo resultado produzido no gerador de malhas deste trabalho. Os valores definidos da equação 2.1 foram: $A = 10,0$, $x_0 = 0,0$, $y_0 = 0,0$, $d_1 = 20,0$ e $d_2 = 20,0$.

Nota-se que a região mais alta, definida pela curva da Figura 4.2 (a) está associada ao maior refinamento de elementos, visto na Figura 4.2 (b).

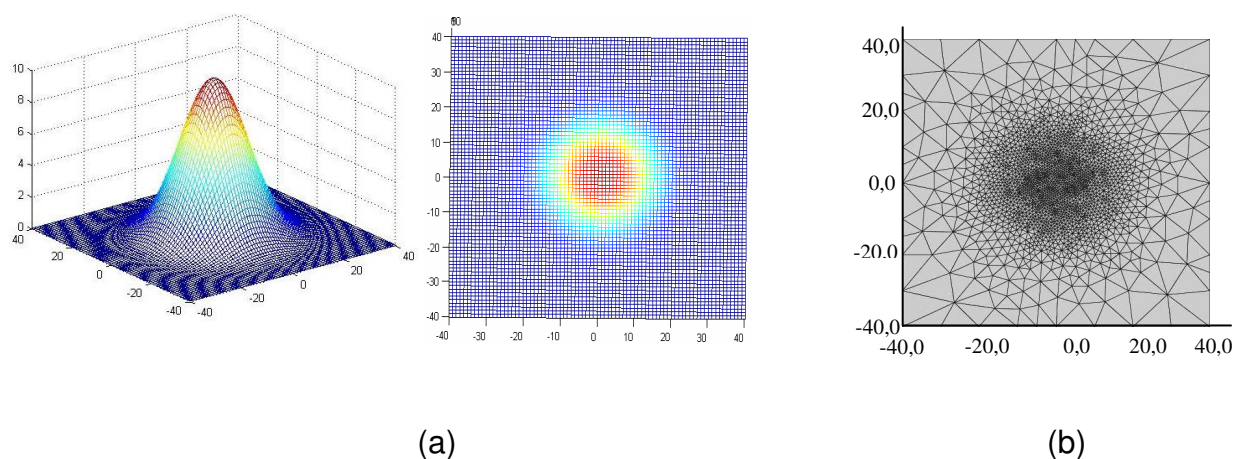


Fig. 4.2: Função gaussiana pré-estabelecida, gerada (a) no Matlab, e no (b) gerador deste projeto.

De forma semelhante, a função de densidade substituída pela equação (2.2), cuja característica de concentração de elementos, encontra-se nos cantos da estrutura, forma um *spot* retangular ao centro. Nesta função, os valores definidos foram: $A = 0,0001$, $a = -10,0$, $b = 10,0$, $c = -10,0$ e $d = 10,0$. A curva dessa função pode ser observada na Figura 4.3 (a), gerada no Matlab, e na Figura 4.3 (b), realizada no gerador deste projeto.

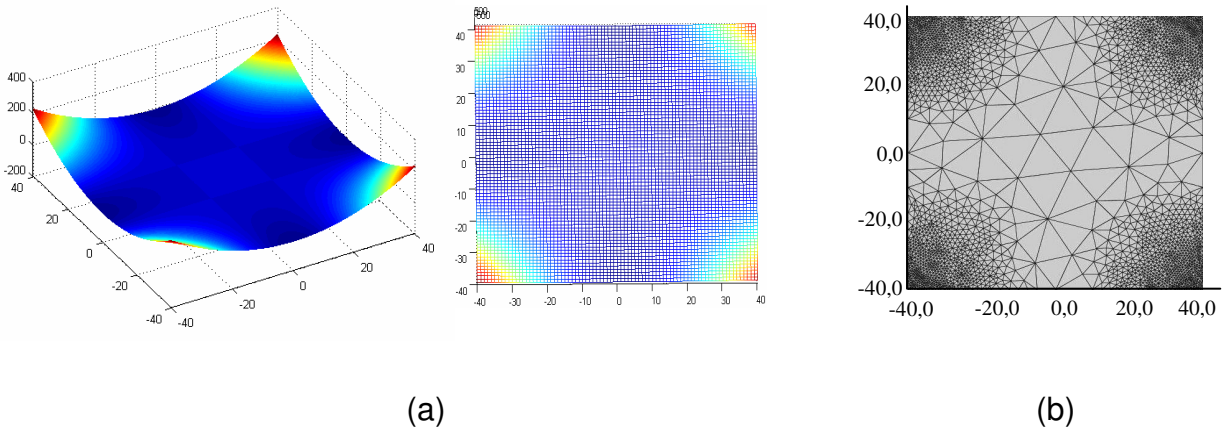


Fig. 4.3: Função retangular pré-estabelecida, gerada (a) no Matlab, e no (b) gerador deste projeto.

Ademais, quando a estrutura apresenta um conjunto complexo de geometrias, é importante possibilitar a formação de várias regiões de alta densidade, como mostrada na Figura 4.4 (a). A Figura 4.4 (b) apresenta uma estrutura com quatro interfaces retangulares internas. Cada interface possui um refinamento denso de elementos no seu centro.

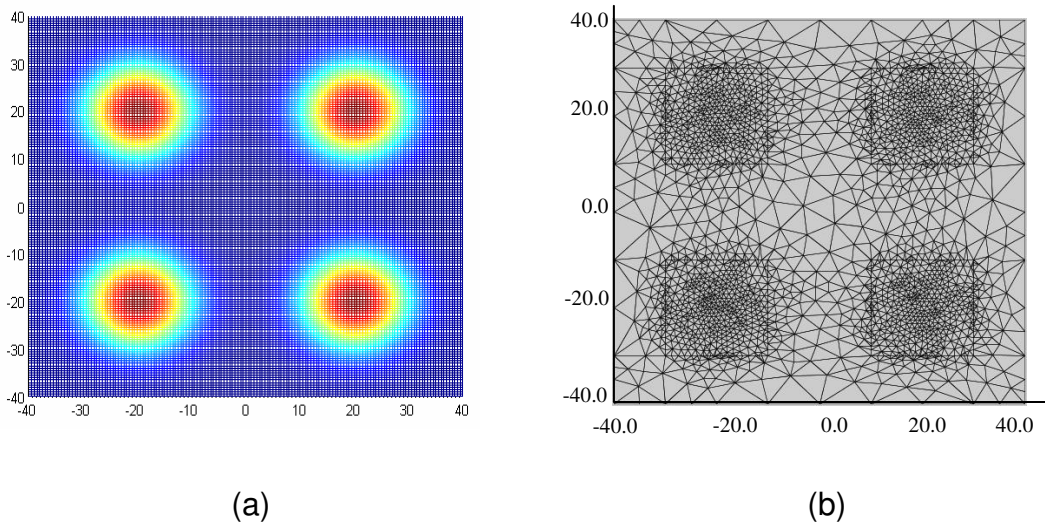


Fig. 4.4: Soma de funções gaussianas, geradas (a) no Matlab, e (b) no gerador deste projeto, ainda com a definição de 4 interfaces internas e retangulares.

A produção dos quatro *spots* deve-se a soma de gaussianas, introduzida pela mesma equação da Figura 4.4 (a). Nesta figura, as posições dos quatro *spots* são: primeiro, $x_0 = -20,0$, $y_0 = 20,0$; segundo, $x_0 = 20,0$, $y_0 = 20,0$; terceiro, $x_0 = 20,0$, $y_0 = -20,0$; e quarto, $x_0 = -20,0$, $y_0 = -20,0$; e tendo como atributos comuns: $A = 1,0$, $d_1 = 10,0$ e $d_2 = 10,0$.

Ainda em relação à função de densidade, de acordo com Fernandez [13], o re-malhamento adaptativo é um método onde uma malha completamente nova é adaptativamente gerada, de acordo com os resultados previamente calculados. Isso permite um remodelamento da malha, obtendo maior precisão na solução sem uma alteração substancial do número de nós produzido. A nova malha é derivada da aproximação da solução anterior, adicionando-se elementos em regiões locais que requerem precisão, e menos elementos onde não são desejados. Ela pode possuir mais, o mesmo ou menor número de elementos que a malha anterior. Um exemplo de remalhamento adaptativo é mostrado na Figura 4.5, que mostra o deslocamento do *spot* da posição $(-20,-20)$ até $(-20;18)$.

As vantagens desse método são evidentes para o caso de uma seqüência de cálculos, ou seja, quando se busca continuamente uma solução, cujo parâmetro esteja sempre variando ou o foco de interesse se mova continuamente no domínio. Isso ocorre, por exemplo, no estudo de guias de ondas ópticos não lineares quando a frequência de onda de entrada sofre mudanças. Assim, o processo de *moving mesh* a que se refere em [13] pode ser obtido através da atualização da função de densidade.

Na próxima seção será descrito detalhadamente o refinamento do elemento, indicada pela linha 6 da Tabela 4.1. O refinamento é explicado através de um exemplo de uma malha inicial, onde é realizada a quebra desses elementos em partes menores, seguido pelas correções de seus aspectos geométricos.

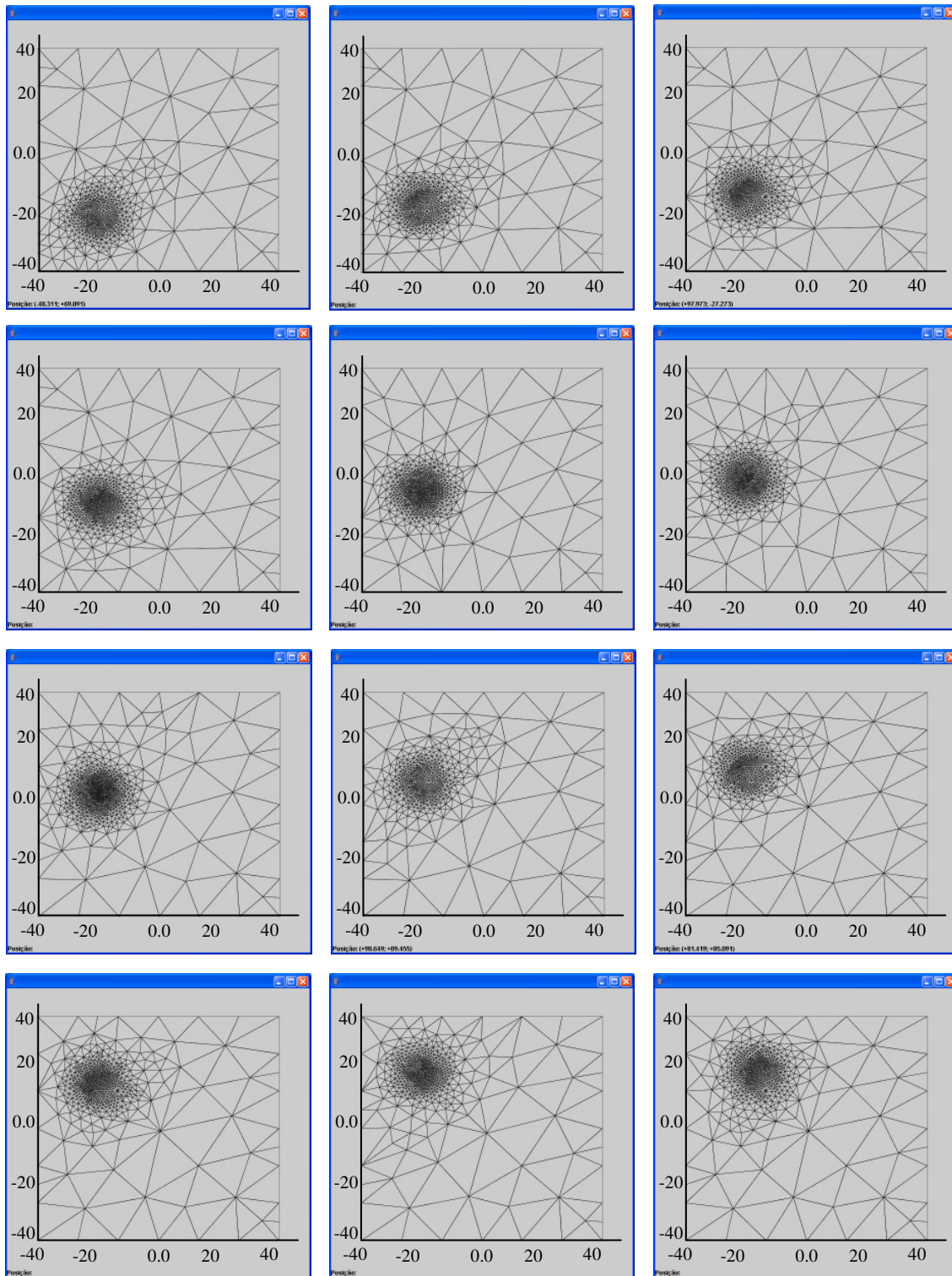


Fig. 4.5: Variação da posição do *spot* em ordem seqüencial.

4.2 Descrição do Método da Decomposição da Topologia

Tendo em vista as técnicas de refinamento apresentadas no Capítulo 3, a descrição particularizada das etapas de refinamento de Delaunay, nesta seção, torna-se mais compreensiva. A desenvolvimento da decomposição da topologia neste trabalho, embora seja semelhante ao algoritmo de inserção incremental de Lawson [10], diferencia-se na maneira como são inseridos os vértices. Baseada na técnica baricêntrica, o nó central é introduzido em todos os elementos, cuja função de densidade, mostrada na Seção 2.3, permita sua subdivisão.

A fim de entender melhor esse processo, segue-se uma breve explicação a ser acompanhada pelas ilustrações da Figura 4.6.

Considere inicialmente uma malha formada por um conjunto de triângulos $T = \{t_1, t_2, t_3\}$, observados na Figura 4.6 (a). Abaixo da malha, o uso de uma representação de contorno quadrado, em volta de cada elemento de T , significa o seu estado de subdivisão. Cada seta, que parte desse contorno em direção à outro, representa um novo elemento gerado ou uma atualização do mesmo. Supõe-se ainda, na Figura 4.6 (a), que a função de densidade $f(x, y)$ considera que o peso individual, de todo elemento de T , seja maior que a densidade por elemento VE . Como consequência, haverá o incremento de um novo vértice central em todos os triângulos de T . Cada novo elemento criado deve ter em comum o nó central como um de seus três vértices.

Na decomposição da malha inicial, visto na Figura 4.6 (a), aplica-se a técnica baricêntrica, produzindo-se o resultado mostrado na Figura 4.6 (b). O novo conjunto de triângulos passará a ser $T = \{t_1, t_2, \dots, t_9\}$. Observa-se que cada contorno quadrado de T produz três novos filhos. Na prática, devem-se inserir no registro dois novos elementos e atualizar o elemento em questão. Por exemplo, na Figura 4.6 (b), no caso do elemento t_1 , atualiza-se o próprio elemento t_1 e criam-se os elementos t_4 e t_5 . E de forma análoga para t_2 e t_3 .

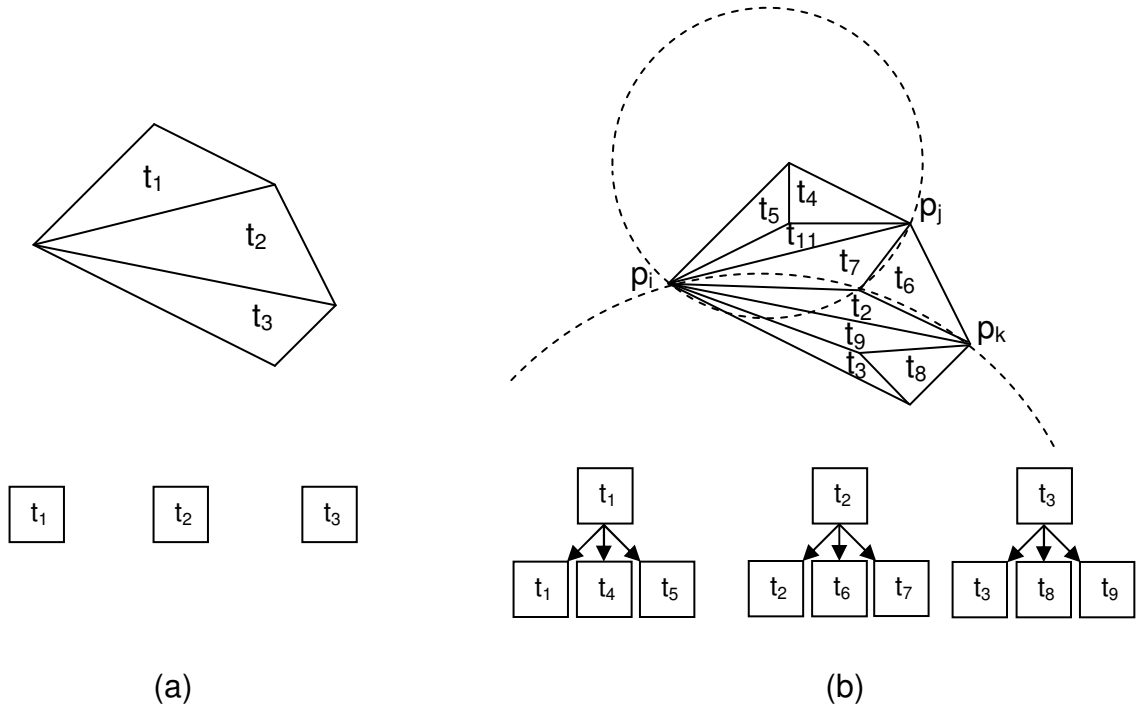


Fig. 4.6: (a) Refinamento de uma malha inicial, aplicando-se a técnica baricêntrica, e após isso, (b) aplicando-se a troca de arestas em $\overline{p_i p_j}$ e, depois, em $\overline{p_i p_k}$.

O processo dá seqüência, ainda na Figura 4.6 (b), com a aplicação do algoritmo de troca de arestas. As trocas ocorrem nas arestas $\overline{p_i p_j}$ e, novamente, em $\overline{p_i p_k}$, que são determinadas através da função *inCircle()*, que considera o critério de Delaunay, descrito na Seção 3.4. A troca de arestas é efetuada analisando-se a formação de um quadrilátero convexo de dois elementos adjacentes, como pôde ser visto nos pares (t_1, t_7) e (t_2, t_9) . A função *inCircle()* pesquisa os três vizinhos do elemento em questão, e verifica o valor de retorno da função *inCircle()*. Caso o valor seja positivo, então, tem-se que a , b e c , são os vértices do elemento em questão e d é o vértice oposto de um elemento adjacente posicionado dentro do circuncírculo. Essa checagem pode ser observada na Figura 4.6 (b), através dos circuncírculos tracejados dos elementos t_2 e t_7 , onde se observa claramente um vértice contido em ambos os circuncírculos. Os detalhes a respeito do algoritmo de detecção de vizinhos e o processo da troca de aresta estão especificados no Apêndice C.

Dessa forma, os elementos t_1 , t_2 , t_7 e t_9 sofrem atualizações, produzindo o resultado mostrado na Figura 4.7 (a). Nesta figura, observa-se que duas setas atingem a mesma representação quadrada. Por exemplo, o novo elemento t_1 é o resultado dos antigos t_1 e t_7 . Isso mostra que a atualização necessita das informações de origem dos dois elementos anteriores.

Em seguida, devido à formação de triângulos de baixa qualidade, especificamente os elementos da interface t_3 , t_4 , t_5 e t_6 , mostrados na Figura 4.7 (a), aplica-se a expansão da técnica baricêntrica, que é a técnica da bissetção pela maior aresta. Essa técnica é efetuada sobre os elementos, onde a maior aresta pertence à interface definida. O resultado da aplicação sobre esses elementos pode ser visto na Figura 4.7 (b). A subdivisão em dois 2 triângulos requisita a atualização destes elementos, e a geração dos novos elementos t_{10} , t_{11} , t_{12} e t_{13} .

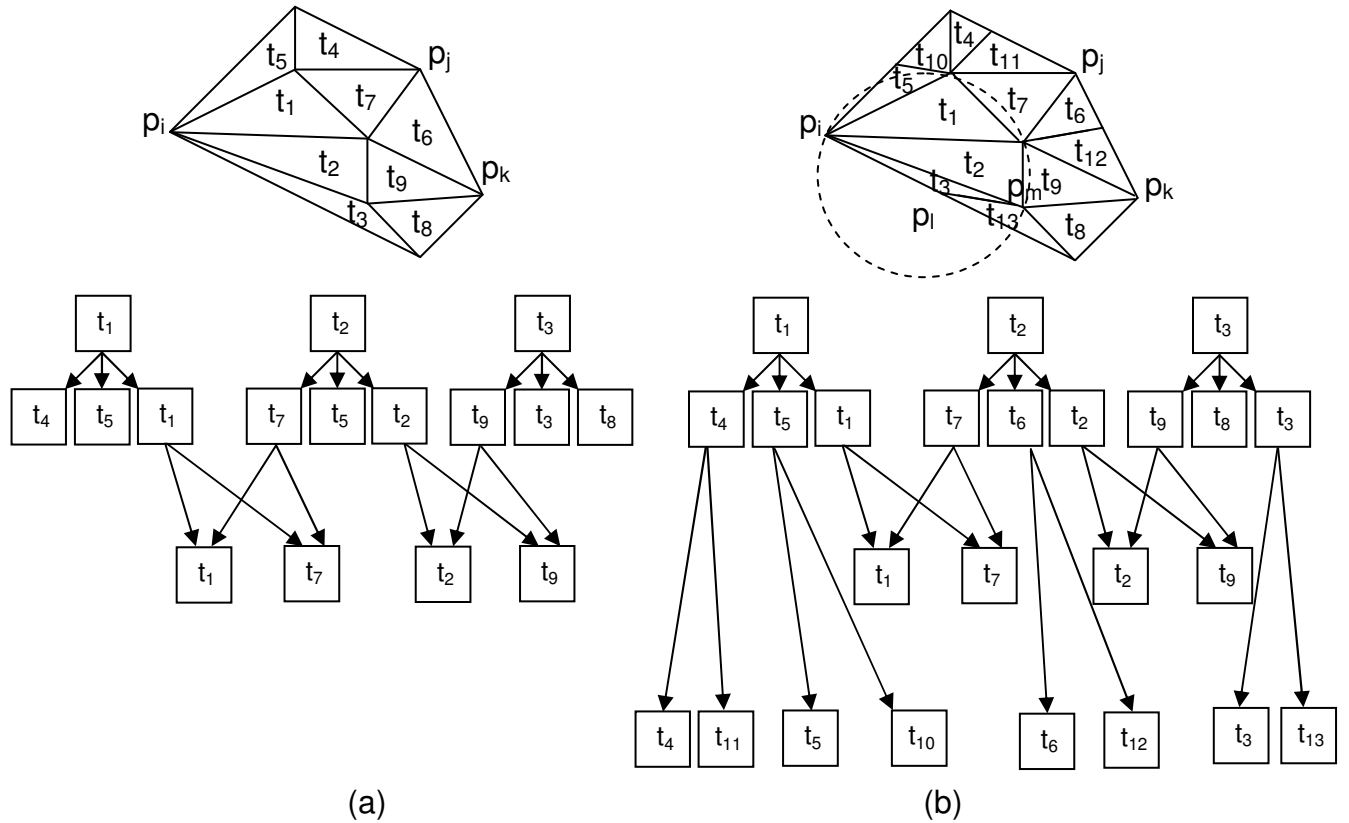


Fig. 4.7: (a) Aplicação da bissetção pela maior aresta, seguida pela (b) troca de arestas em $\overline{p_i p_m}$.

Procedendo-se, a troca de arestas ocorre em $\overline{p_i p_m}$, mostrada na Figura 4.7 (b), que detecta um vértice dentro do circuncírculo do elemento t_2 . Dessa forma, devem-se atualizar os elementos t_2 e t_3 . O resultado pode ser visto na Figura 4.8.

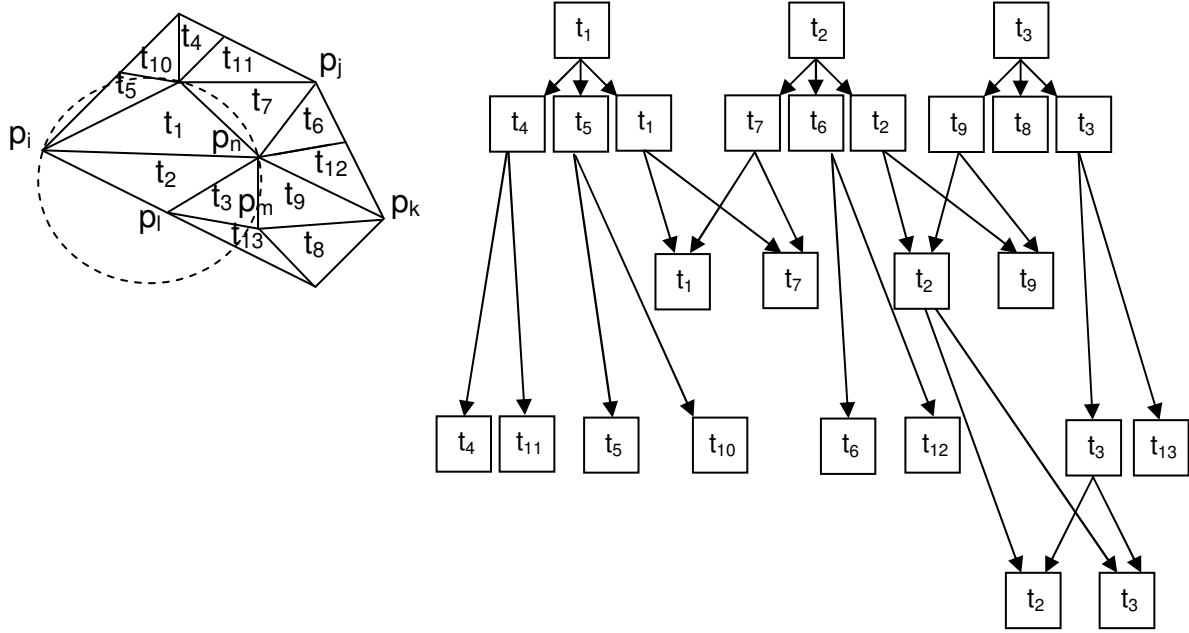


Fig. 4.8: Aplicação da troca de arestas em $\overline{p_i p_n}$.

E, finalmente, termina-se com a troca de arestas em $\overline{p_i p_n}$, mostrada na Figura 4.8, que atualiza os elementos em t_1 e t_2 . O resultado disso pode ser observado na Figura 4.9.

A representação final, gerada pela Figura 4.9, mostra o histórico de todos os treze elementos gerados, podendo-se fazer um acompanhamento retrógrado e observar as suas origens. Observa-se também, as técnicas, identificadas por $T1$, $T2$, $T3$, $T4$ e $T5$, aplicadas sobre a malha durante um ciclo.

A malha final significa um ciclo completo de etapas, ou seja, a próxima iteração aplicará novamente as técnicas mostradas desde a Figura 4.6 até a Figura 4.9. Cada ciclo completo é identificado através da $malha_m$, onde m é incrementado a cada iteração.

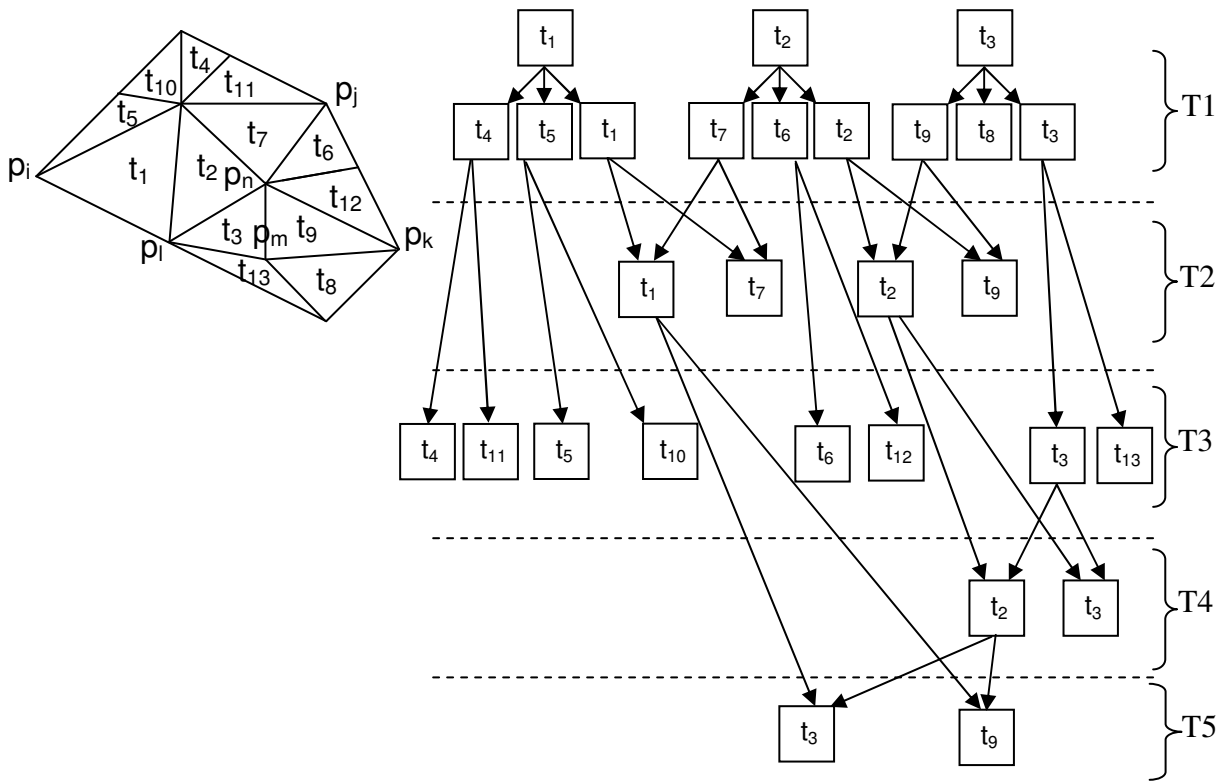


Fig. 4.9: Malha final obtida e o hitórico de todos os elementos gerados.

Em suma, a designação do método da decomposição da topologia é uma combinação de técnicas de refinamento, que parte de uma triangulação inicial. A partir disso, o algoritmo gera um número estimado de elementos, de tal forma que cubra todo o domínio, após sucessivas subdivisões.

Nota-se que os primeiros elementos subdivididos dependem da topologia inicial da malha, que é uma característica da técnica baricêntrica, onde os resultados produzem elementos distorcidos. Para contornar essa situação, os elementos devem passar por duas técnicas, a de Delaunay e a da bissecção. Na próxima seção será descrito como a técnica da bissecção garante a integridade das interfaces geométricas definidas.

4.3 Integridade das Interfaces Geométricas

O gerador de elementos deve ser capaz de garantir a integridade das interfaces. Durante o refinamento da malha criam-se segmentos discretos menores, os quais formarão a interface inicialmente definida. Assim, quanto maior o número de segmentos discretos, maior será a definição desta interface. Entretanto, o maior número de segmentos pode causar degradações nas interfaces, se não houver uma solução satisfatória de integridade de bordas, durante a geração de malhas.

Para evitar a degradação do contorno, é necessária a aplicação de uma técnica de ajustes nas interfaces. Trabalhos como [15] e [25], que tratam sobre a geração automática de malhas utilizando a técnica de Delaunay, mostram a recuperação de uma interface previamente deteriorada, como a descrita na Figura 3.10, da Seção 3.2. Em seguida, a mesma é recuperada através da inserção de um nó, denominado ponto de recuperação (*stitch point*), que deve se localizar sobre a interface. Assim, o novo conjunto de arestas, que interligam ao ponto de recuperação, restabelece a interface geométrica do domínio. Caso o cálculo do ponto de recuperação não seja feito corretamente, a interface não será corretamente formada ao longo das iterações. Então, ocorrerá a formação de buracos topológicos, que são malhas com interfaces degeneradas.

Neste projeto, o cálculo do ponto de recuperação está diretamente associado à técnica da bissecção aplicada nas interfaces.

A seguir, é realizada uma descrição sobre a localização do ponto de recuperação. Este, deverá se localizar sobre as diversas interfaces instituídas, tais como retilínea, retangular, circular, elíptica e curva arbitrária.

4.3.1 Interface Retangular

A interface retangular é constituída de segmentos retilíneos que delimitam o seu domínio. A integridade deste tipo de interface, após sucessivas subdivisões,

será garantida se for determinada a localização correta do ponto de recuperação numa aresta de triângulo pertencente à interface.

Embora seja relativamente simples, devem-se considerar alguns tratamentos a fim de se evitar os buracos topológicos. É relevante classificar os elementos de duas formas: caso o elemento da interface possua um par oposto ou não. O significado do par oposto associa-se ao elemento adjacente que compartilha a mesma aresta da interface. Caso não exista, basta inserir um novo vértice no ponto médio $p'(x', y') = \left(\frac{x_2 + x_3}{2}, \frac{y_2 + y_3}{2} \right)$ da aresta e subdividir o elemento, conforme mostrado na Figura 4.10 (a). Caso exista, devem-se pesquisar quais dos três vizinhos adjacentes possui a mesma aresta em comum e, dessa forma, manter a conformidade na subdivisão com os pares opostos, como visto na Figura 4.10 (b), onde os pontos p_1 , p_2 e p_3 são os três vértices que compõem o elemento em questão; po_1 , po_2 e po_3 são os três vértices que compõem o par oposto; e $p' = (x', y')$ é o novo vértice adicionado que pertence à fronteira. Este algoritmo de integridade pode ser visto no Apêndice C.

Salienta-se que os pontos $p_1, p_2, p_3, po_1, po_2$ e po_3 são apenas critérios de identificações das Figuras 4.10 (a) e (b), pois na realidade não existem dois nós globais com a mesma coordenada, portanto, os nós po_3 e p_2 associam-se a um mesmo nó global; da mesma forma que os nós po_2 e p_3 .

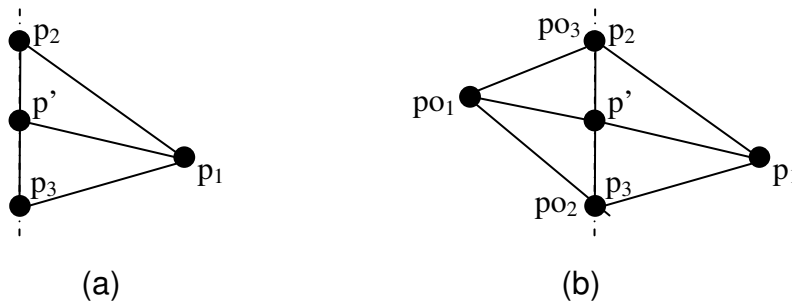


Fig. 4.10: A localização do novo nó $p' = (x', y')$ num (a) elemento de interface externa, e num (b) elemento de interface interna.

A aplicação dessa técnica permite, portanto, realizar o refinamento nos elementos de interfaces retilíneas. Alguns exemplos de geração de malhas obtidos, pelos algoritmos desenvolvidos neste trabalho, podem ser vistos na Figura 4.11 (a) e (b). Na primeira imagem, tem-se um exemplo de geração de malha que possui uma interface geométrica definida em formato “L”. E, na segunda imagem, tem-se a inserção de uma geometria em formato “H”. A geração de elementos de ambas as malhas foram estimadas em 1500.

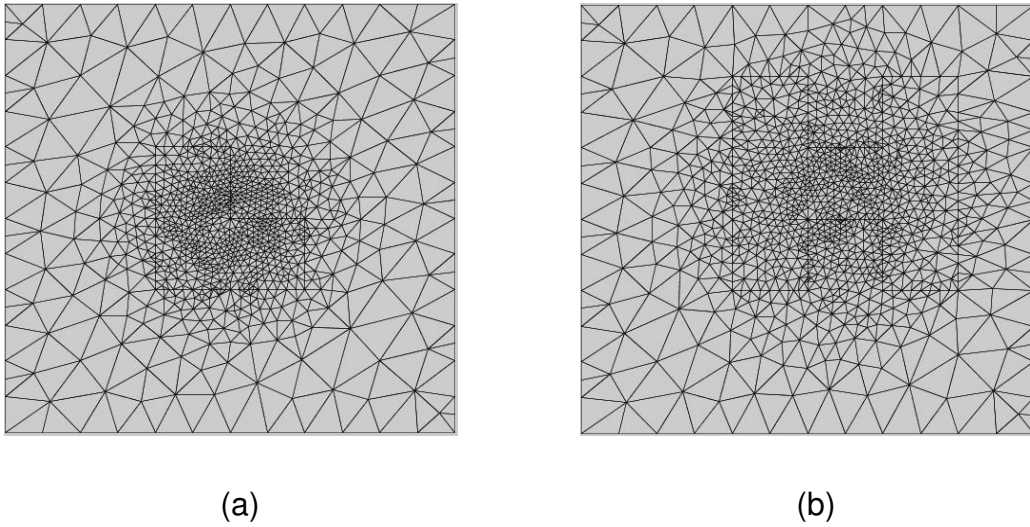


Fig. 4.11: Exemplo de geração de malha para uma geometria em (a) formato “L”, e em (b) formato “H”.

4.3.2 Interface Circular

A garantia de integridade das interfaces circulares pode ser realizada conhecendo-se a sua representação paramétrica. Na figura 4.12 (a), considera-se uma circunferência de raio a e centro $C_0 = (x_0, y_0)$, onde V_n , para $n = 1, \dots, 4$, são os nós da malha; θ_A e θ_B correspondem, respectivamente, aos ângulos AC_0V_1 e AC_0V_3 .

Inicialmente, ao se levar em conta um sistema (x', y') , é importante considerar a translação de C_0 para a origem O , que é dada por $x' = x - x_0$ e

$y' = y - y_0$, para facilitar os cálculos. O cálculo do ponto de recuperação, a partir da Figura 4.12 (a), é dado pelo ponto localizado no arco médio de V_1V_3 , cujo ângulo pode ser obtido por $\theta = (\theta_A + \theta_B)/2$.

Obtendo-se a informação de θ e conhecendo-se a representação paramétrica de uma circunferência, que é dada por $x' = a \cdot \cos \theta$ e $y' = a \cdot \sin \theta$, calcula-se o ponto de recuperação. Esse ponto é dado por $V_5 = (x'_5, y'_5)$, mostrado na Figura 4.12 (b), e associado ao ângulo AC_0V_5 . O resultado após uma subdivisão é mostrado na Figura 4.12 (b), onde V_n , para $n=1, \dots, 5$, é o novo conjunto de nós.

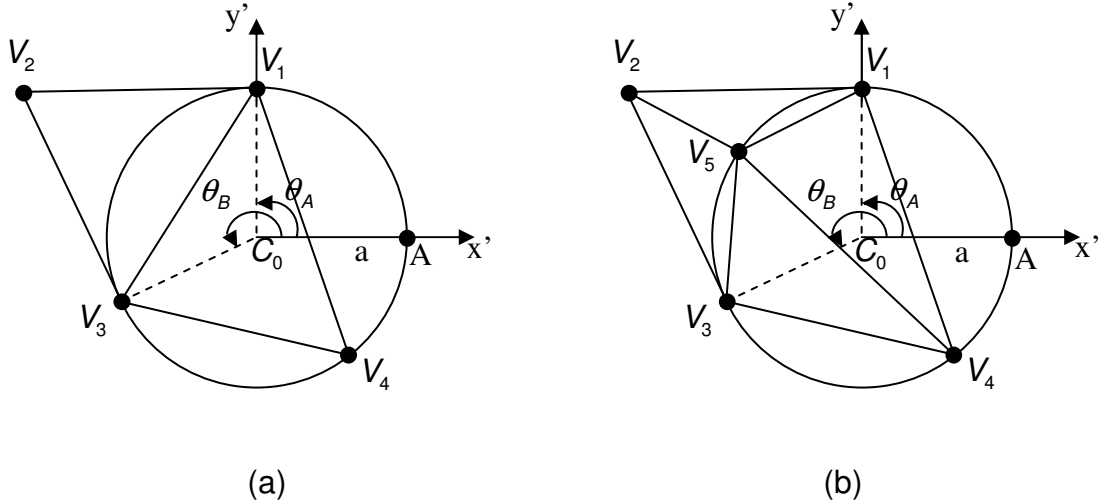


Fig. 4.12: Integridade da interface circular, a partir de uma (a) disposição inicial, e após a (b) inserção do ponto de recuperação.

Portanto, a inserção do ponto de recuperação dentro do quadrilátero composto pelos pontos V_1 , V_2 , V_3 e V_4 gera dois novos elementos (V_5, V_2, V_3) e (V_5, V_3, V_4) ; e a atualização dos elementos antigos (V_1, V_2, V_3) e (V_1, V_3, V_4) , respectivamente, por (V_5, V_1, V_2) e (V_5, V_4, V_1) .

As Figuras 4.13 (a) e (b) mostram exemplos de resultados obtidos, onde a função de densidade localiza-se no centro da interface circular. A primeira figura possui 818 elementos, e a segunda, malha mais densa, possui 4632 elementos.

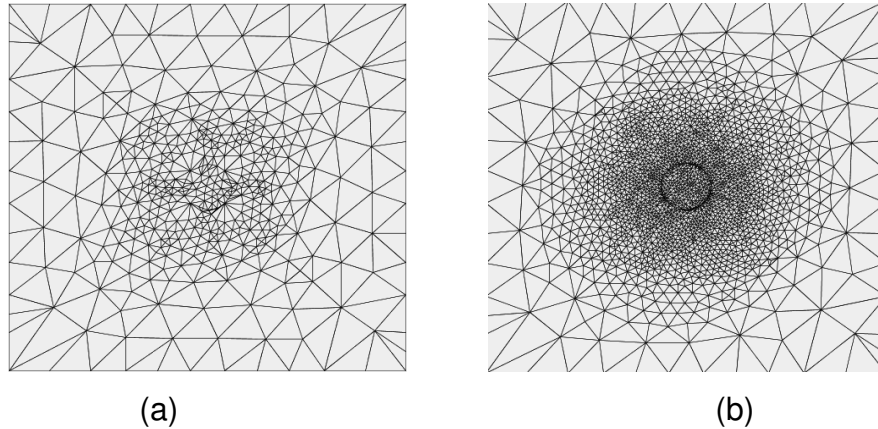


Fig. 4.13: Exemplo de geração de malha, com (a) 818 elementos, e (b) 4632 elementos.

Os detalhes a respeito do cálculo do ponto de recuperação, associado ao algoritmo de interface circular e a sua descrição detalhada, são mostrados no Apêndice C.

O processo de implementação é semelhante para elipses, e neste caso devem-se considerar os raios vertical e horizontal da representação paramétrica.

4.3.3 Interface de Curvas Arbitrárias

Trabalhos que expõem a prática de geração de malhas em curvas arbitrárias são relativamente difíceis de encontrar. Neste estudo, chegou-se a uma solução, quanto à localização do ponto de recuperação, através de uma curva paramétrica.

Para se compreender a associação de uma malha à curva de Bézier, técnica computacional empregada e que pode ser obtido em [1], é necessária a introdução de alguns conceitos. A introdução de uma curva linear, na qual dados os pontos P_0 e P_1 , forma uma linha reta que liga os mesmos. A representação desta interpolação linear é vista em (4.1),

$$B(t) = P_0 + (P_1 - P_0)t = (1 - t)P_0 + tP_1 \quad (4.1)$$

onde $t \in [0,1]$.

Da mesma forma, uma curva quadrática é uma curva traçada pela função $B(t)$, formada pelos pontos P_0 , P_1 e P_2 , e mostrada em (4.2). E uma curva cúbica, formado pelos pontos P_0 , P_1 , P_2 e P_3 , pode ser vista em (4.3),

$$B(t) = (1-t)^2 P_0 + 2t(1-t) P_1 + t^2 P_2 \quad (4.2)$$

$$B(t) = (1-t)^3 P_0 + 3P_1 \cdot t(1-t)^2 + 3P_2 \cdot t^2(1-t) + P_3 t^3 \quad (4.3)$$

onde $t \in [0,1]$.

Portanto, a generalização de uma curva de Bézier de grau n , dado os pontos $P_0 \dots P_{n-1}$, é dada por (4.4),

$$B(t) = \sum_{i=0}^n \binom{n}{i} P_i (1-t)^{n-i} \cdot t^i = P_0(1-t)^n + \binom{n}{1} P_1 (1-t)^{n-1} \cdot t + \dots + P_n \cdot t^n \quad (4.4)$$

onde $t \in [0,1]$ e n define o grau da curva.

A sua terminologia pode ser vista em (4.5),

$$B(t) = \sum_{i=0}^n P_i B_{n,i}(t) \quad (4.5)$$

onde $t \in [0,1]$ e $B_{n,i}(t) = \binom{n}{i} \cdot t^i (1-t)^{n-i}$, para $i = 0, \dots, n$. Esta última expressão (4.5) é conhecida por base polinomial de Bernstein [1].

Dá-se seqüência, associando-se a curva à construção de malhas. Na prática, os pontos de controle P_0, \dots, P_{n-1} provêm apenas informações direcionais da curva. Dessa forma, deve-se ter conhecimento da distinção entre os pontos de controle P_0, \dots, P_{n-1} e os nós p_0, \dots, p_m gerados sobre a curva arbitrária, que será apresentada a seguir.

Um esboço de curva, mostrada na Figura 4.14 (a), é criado através da definição dos pontos de controle P_0, \dots, P_5 . Como pode ser observado na Figura 4.14 (b), os pontos gerados pertencentes à curva são os pontos p_0, \dots, p_9 , embora os pontos de controle P_0 e P_5 , também façam parte, pois são os pontos extremos da curva. O parâmetro t varia de zero a um, respectivamente, do início ao fim da

curva. Nesta, estipulou-se, na variação dos pontos, um intervalo de 0,1 no parâmetro t de (4.5).

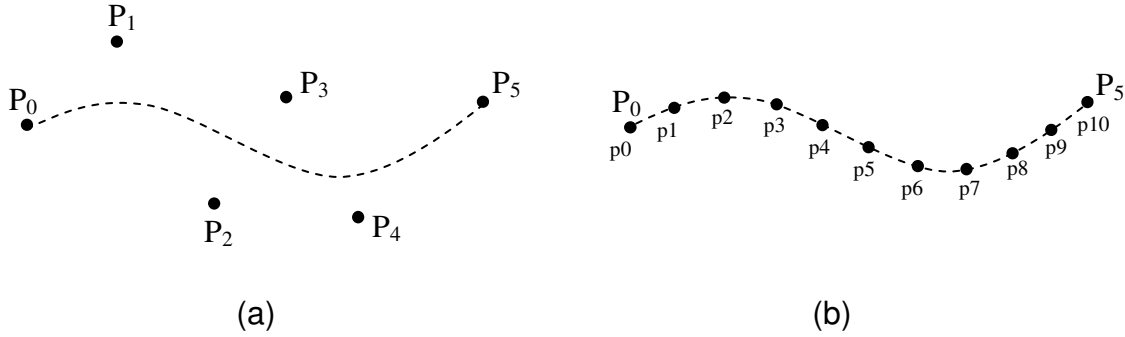


Fig. 4.14: Definição de uma curva arbitrária através dos (a) pontos de controle P_0, \dots, P_5 , e a (b) geração dos nós p_0, \dots, p_9 da curva.

A Figura 4.15 (a) mostra a mesma curva, com os valores de t variando em 0,25. Na primeira parte da curva, com o parâmetro t variando de 0,0 a 0,5, existem dois tipos de elementos: o elemento superior e o inferior. A Figura 4.15 (b) mostra parte da curva, cujo parâmetro varia de 0 até 0,5, e com o elemento superior (p, p_0, p_1) . Sabe-se que o elemento pertencente à curva deve possuir dois vértices sobre ela, e está representado pelos pontos p_0 e p_1 . Esse elemento deve armazenar um parâmetro t_{\min} em p_0 , e t_{\max} em p_1 . Os parâmetros t_{\min} e t_{\max} indicam a localização dos dois vértices do elemento sobre a curva. Assim, em relação ao elemento mostrado na Figura 4.15 (b), tendo o $t_{\min} = 0$ e $t_{\max} = 0,5$, o parâmetro médio deve-se situar em $t_{\min} < t_{med} < t_{\max}$, ou, mais precisamente, na média dos extremos, que resulta em 0,25. Conhecendo-se o valor de t_{med} , e efetuando o cálculo visto em (4.5), descobre-se a coordenada do novo nó $B(t_{med})$ sobre a curva. Esse novo ponto corresponde ao ponto de recuperação que permitirá realizar a nova subdivisão do elemento correspondente. A subdivisão em três novos elementos, observada na Figura 4.15 (c), está representada pela linha pontilhada, após a inserção do ponto p' .

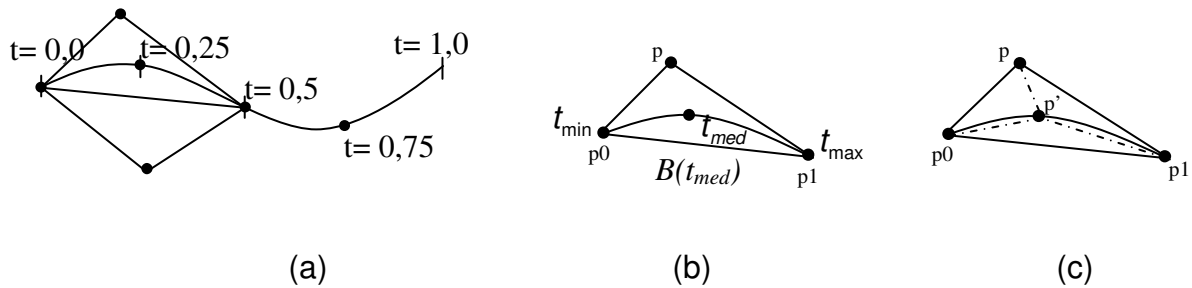


Fig. 4.15: Curva arbitrária, com (a) variação de 0,25 no parâmetro t ; o (b) elemento superior com parâmetros t_{\min} , t_{med} e t_{\max} ; a (c) inserção de p' sobre a curva.

Em outro procedimento, tem-se, como exemplo na Figura 4.16 (a), uma estrutura com interface externa em forma de pentágono e uma interface interna com curvatura. A curva na região central é definida através das informações dos pontos de controle fornecidos por P_0 , P_1 e P_2 , mostrados na Figura 4.16 (b). Nesta figura, particularmente, os pontos gerados da curva derivam de uma variação incremental de 0,05 para o parâmetro t . O ponto para $B(0,5)$ corresponde ao ponto mais alto da curva.

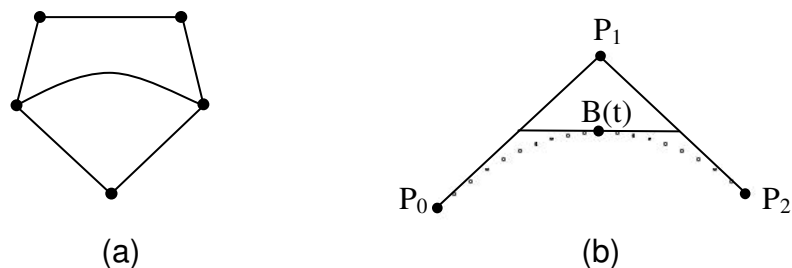


Fig. 4.16: Estrutura com uma (a) curva arbitrária simétrica interna na região central; a (b) definição dos pontos de controle, e posteriormente, a geração de nós da curva com variação de $t = 0,5$ no parâmetro.

A Figura 4.17 (a) corresponde à malha inicial utilizada para o refinamento. Conforme visto na Figura 4.17 (b), nessa configuração inicial, existem dois

intervalos do tipo $t_{\min} < t < t_{\max}$: o intervalo₁ $0 < t < 0,5$ e o intervalo₂ $0,5 < t < 1$. Esses intervalos encontram-se armazenados nos atributos dos elementos pertencentes à interface, possibilitando a próxima quebra. Na primeira quebra, o valor de t_{med} para o primeiro e segundo intervalo será, respectivamente, 0,25 e 0,75. E assim sucessivamente. Um exemplo de processo iterativo de quebra é mostrado na Figura 4.17 (c).

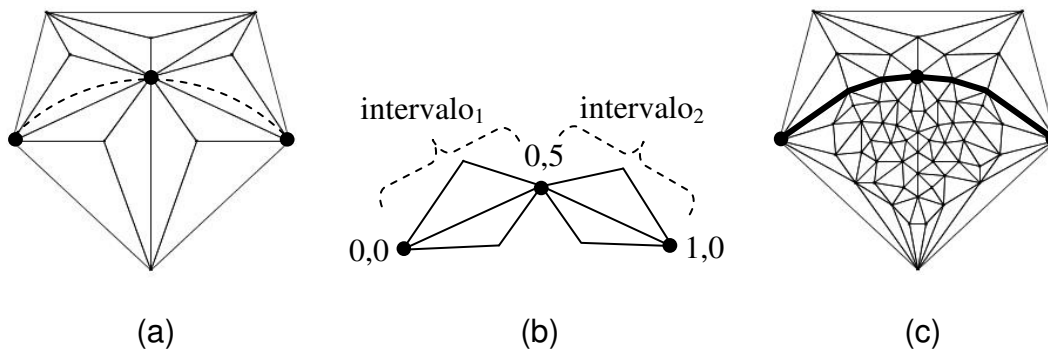


Fig. 4.17: Geração de uma malha com uma curva simétrica, a partir de uma (a) malha inicial; a (b) divisão em 2 intervalos, e o (c) processo iterativo de quebra.

O próximo teste é realizar uma distorção da curva através do deslocamento do ponto de controle P_1 , mostrado na Figura 4.18 (a). A curva apresentada tem uma variação incremental de 0,05 para o parâmetro t . A malha resultante, na Figura 4.18 (b), deve produzir um acompanhamento da curva.

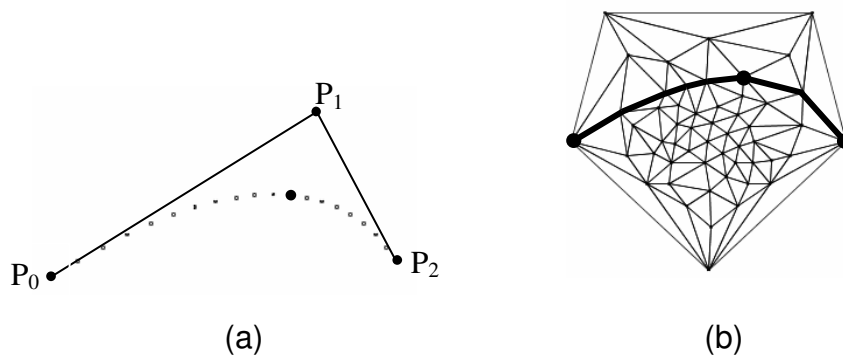


Fig. 4.18: (a) Curva arbitrária assimétrica com a definição dos pontos de controle; e o (b) processo iterativo de quebra.

Finalmente, um exemplo de curva fechada pode ser visto na Figura 4.19. Na Figura 4.19 (a), tem-se o esboço de uma curva arbitrária interna a uma interface quadrada.

Na Figura 4.19 (b), tem-se a construção da curva através da definição de nove pontos de controle com variação incremental de 0,05 em t . Observa-se também, por opção do autor, a definição de quatro intervalos: $0,0 \leq \text{intervalo}_1 \leq 0,25$, $0,25 < \text{intervalo}_2 \leq 0,5$, $0,5 < \text{intervalo}_3 \leq 0,75$ e $0,75 < \text{intervalo}_4 \leq 1,0$. Para uma curva fechada, o número mínimo de intervalos deve ser três, e para curvas abertas, um único intervalo já é suficiente.

Na Figura 4.19 (c), tem-se a utilização de uma malha inicial qualquer, construída a partir da definição do número de intervalos.

Nota-se que o nó inicial da curva corresponde ao próprio nó final da mesma. Dessa forma, os parâmetros $t = 0,0$ e $t = 1,0$ associam-se ao mesmo nó. O processo de implementação é semelhante ao que vimos anteriormente.

Como resultado, tem-se que na Figura 4.20 (a) produziu-se 660 elementos e 335 nós, e a sua malha mais densa, mostrada na Figura 4.20 (b), possui 1190 elementos e 600 nós.

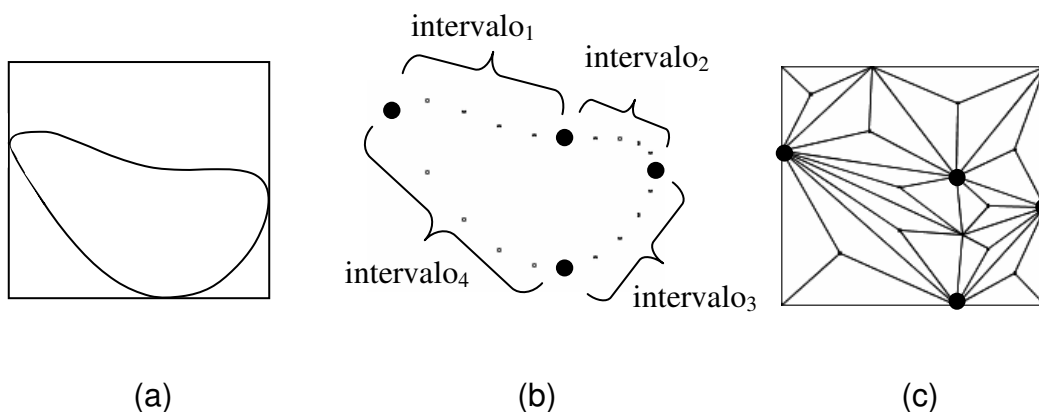
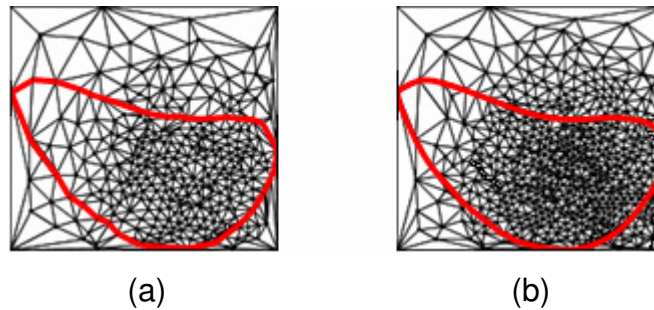


Fig. 4.19: Geração de malha partindo-se de uma (a) estrutura de curva arbitrária fechada, e os (b) nós gerados pertencentes à mesma com a definição de quatro intervalos. A partir dos intervalos definidos, monta-se uma (c) malha inicial para o refinamento.



4.20: Resultados de uma malha refinada com (a) 660 elementos, e com (b) 1190 elementos.

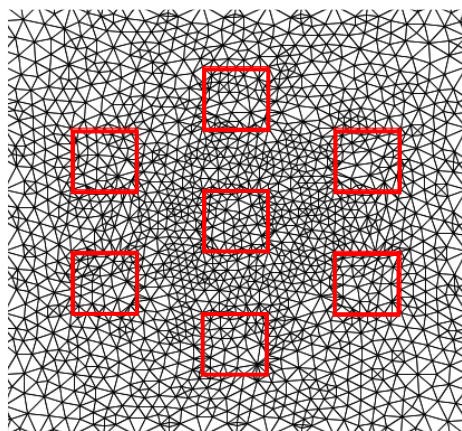
O algoritmo de integridade de interface de curvas arbitrárias e sua descrição detalhada são mostrados no Apêndice C.2.3.

4.3.4 Exemplos de Estruturas

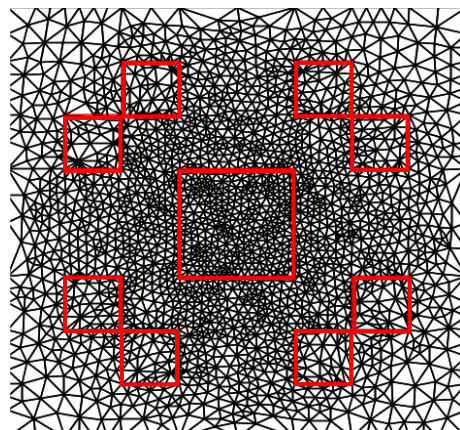
Nas Figuras 4.21, 4.22 e 4.23 apresentam-se alguns exemplos de malhas, com interfaces retangulares e circulares, de acordo com os procedimentos dos algoritmos de integridade de bordas mencionados anteriormente.

A Figura 4.21 mostra exemplos de estruturas retangulares, na região superior; e circular com uma visão ampliada desta interface, na região inferior.

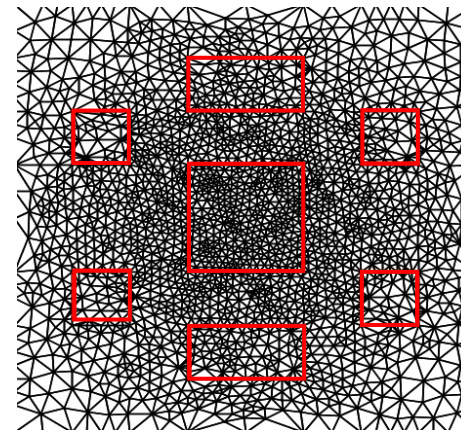
As Figuras 4.22 e 4.23 mostram exemplos de estruturas de interfaces circulares com variações de tamanhos.



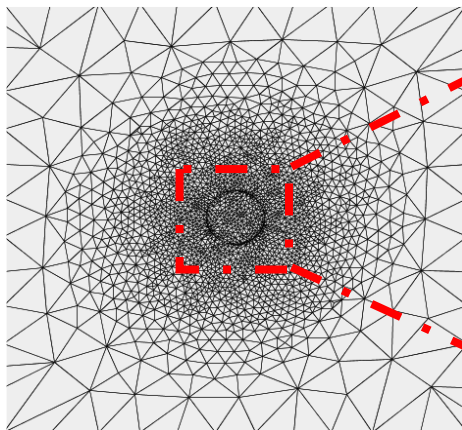
3720 elementos
1909 nós



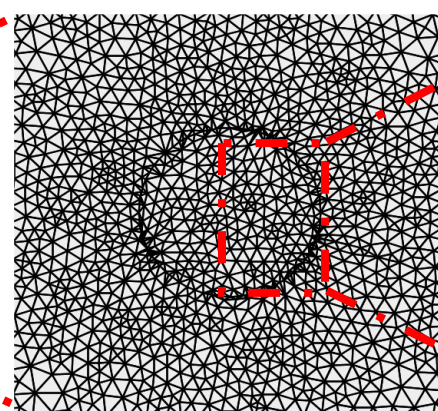
4812 elementos
2448 nós



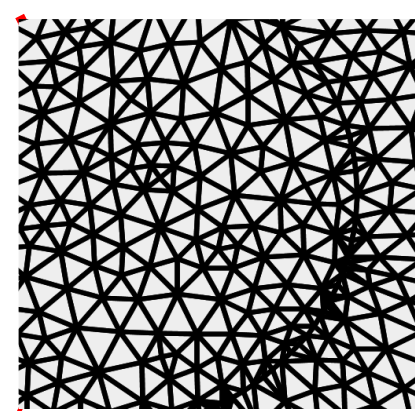
4696 elementos
2390 nós



4560 elementos
2306 nós

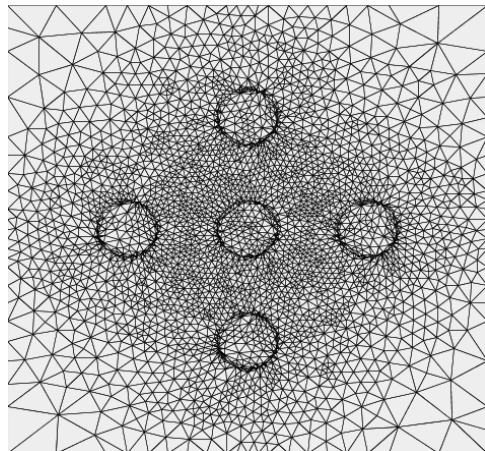


Visão ampliada em 3 vezes

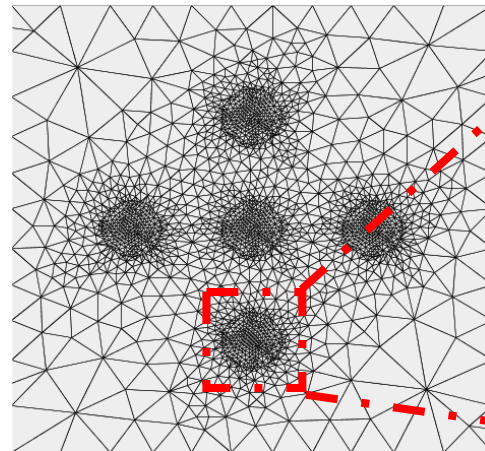


Visão ampliada em 10 vezes

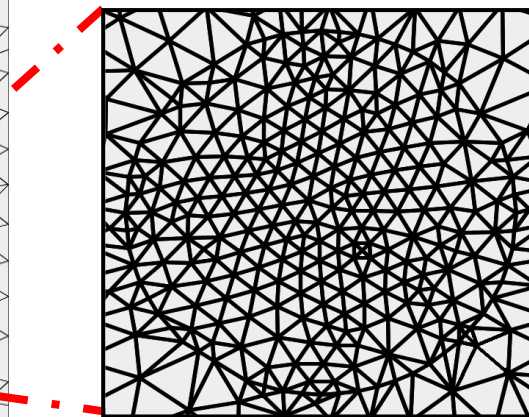
Fig. 4.21: Exemplos de geração de malhas.



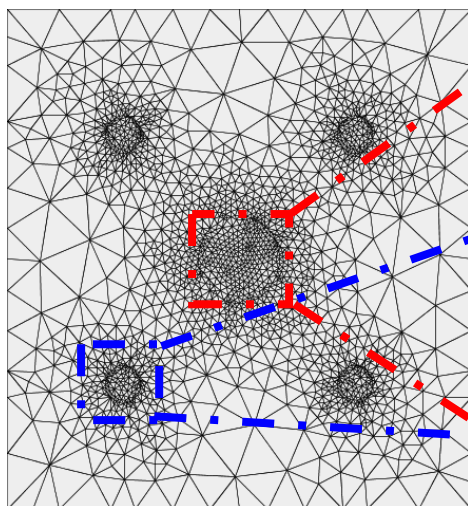
6728 elementos
3408 nós



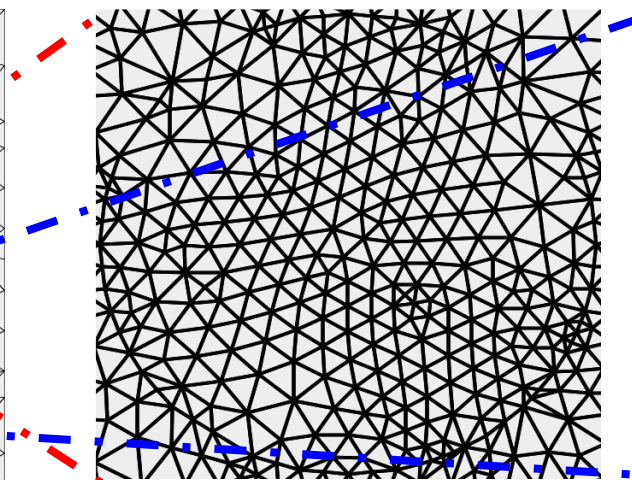
4304 elementos
2178 nós



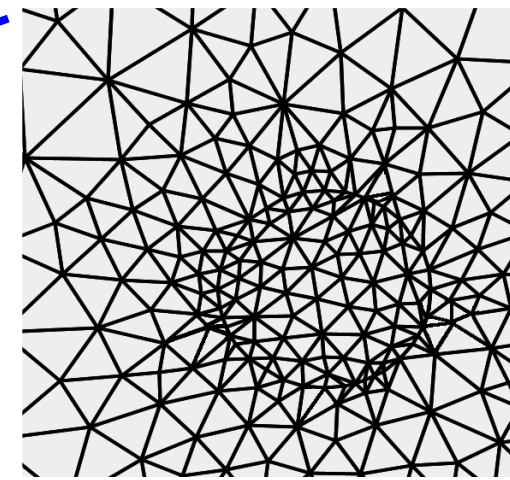
Visão ampliada em 8 vezes



3696 elementos
1876 nós

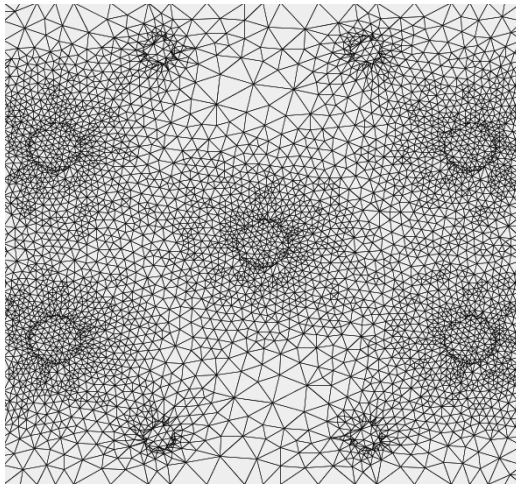


Visão ampliada em 8 vezes



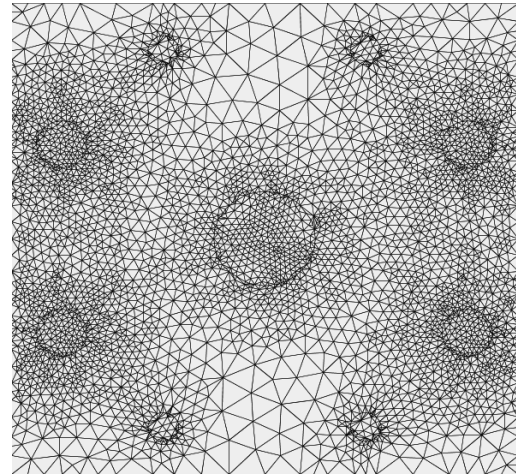
Visão ampliada em 8 vezes

Fig. 4.22: Exemplos de geração de malhas com interfaces circulares.



8376 elementos

4270 nós



8306 elementos

4236 nós

Fig. 4.23: Geração de malhas em estruturas complexas.

Capítulo 5

Estudos Comparativos

Este capítulo realiza um estudo comparativo entre os exemplos de malhas obtidas no gerador e as referências nas teses de doutorado de Baude [2] e Ribeiro Filho [5], que será relevante para uma análise gráfica posterior dos aspectos qualitativos de uma malha. O cálculo de qualidade utilizado na avaliação é o fator α , apresentado na Seção 3.4, devido à sua característica conservadora de classificação e também como um indicador qualitativo utilizado em outras publicações como [11] e [34].

Apresenta-se também um estudo avançado na flexibilidade da geração de malhas, alterando-se o número de nós de contorno. Esta alteração gera distribuições diferentes dos elementos após o refinamento; ainda possibilita realizar uma comparação entre malhas distintas das mesmas estruturas; e também permite realizar trabalhos futuros na implementação de técnicas de refinamento, por exemplo, partindo-se da definição do tamanho das arestas dos elementos pertencentes às interfaces, que é um critério utilizado no GiD.

5.1 Comparação entre Técnicas de Suavização

Nesta seção efetua-se uma comparação entre as duas técnicas de suavização que são: a suavização laplaciana e a sua variação restrita, onde estes algoritmos já foram apresentados, respectivamente, na Tabela 3.4 e 3.5, da Seção

3.7. Como muitas vezes o cálculo do fator de qualidade fornecido pelo Delaunay ainda não fornece resultados satisfatórios, esse estudo mostrará uma mudança do aspecto visual da malha, com a aplicação da suavização, após a triangulação de Delaunay.

A Figura 5.1 mostra as iterações da suavização laplaciana, tendo como malha original a Figura 5.1 (a), que foi obtida após a técnica de Delaunay. Após uma iteração da suavização laplaciana, mostrada na Figura 5.1 (b), pode-se notar o início do processo de equilíbrio da malha. As Figuras 5.1 (c) e (d) mostram, respectivamente, o que ocorre após duas e três iterações.

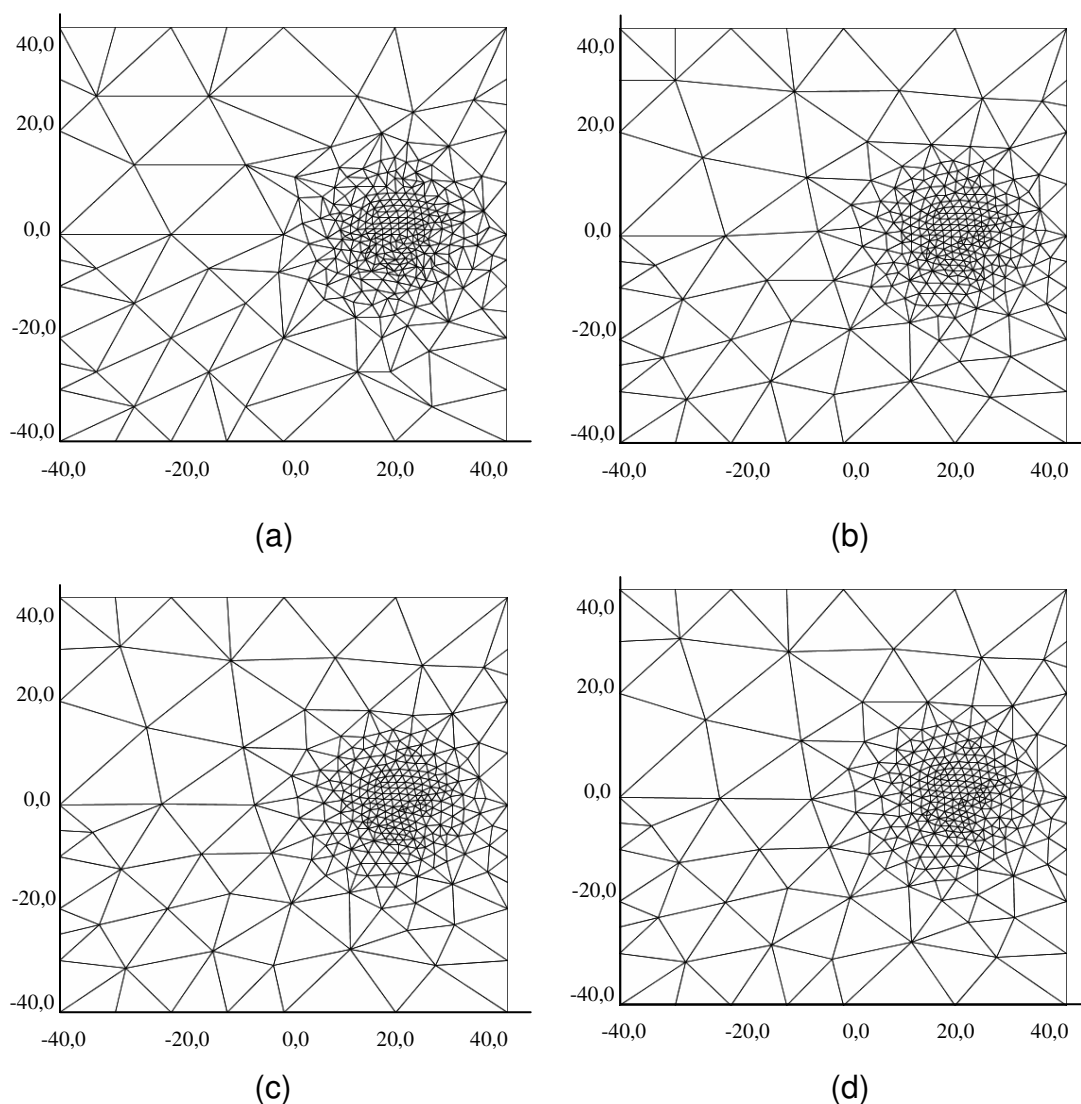


Fig. 5.1: Malha obtida após a (a) técnica de Delaunay, aplicando-se a suavização laplaciana com (b) uma iteração, (c) duas iterações, e (d) três iterações.

Através da seqüência de processos mostrados na Figura 5.1, nota-se a melhoria do aspecto visual da malha, que neste exemplo específico, também deve concorrer com a melhoria da qualidade individual dos elementos.

O maior número de iterações não significa maior mudança significativa no resultado da malha, quando já se obtém o equilíbrio dos nós. No trabalho de Baude [2], por exemplo, definiu-se uma suavização recursiva até que nenhum deslocamento significativo (em torno de 0,1 mm) na posição dos nós ocorresse. Outros trabalhos como de Ribeiro Filho [5] definem uma quantidade fixa $N_{iteração}$ de iteração.

A seguir, para distinguir melhor o efeito produzido pela técnica de Delaunay e as suavizações laplaciana e a laplaciana restrita, comparou-se o resultado de três tipos de malhas.

A Figura 5.2 (a) foi obtida após a aplicação da técnica de Delaunay, onde se percebe claramente a existência entre elementos de transições abruptas.

O histograma da Figura 5.2 (b) mostra o fator α em função da probabilidade de ocorrência dos elementos normalizados. Nele, nota-se uma concentração significativa de elementos distribuídos em $0,8 \leq \alpha \leq 1,0$; e para $\alpha < 0,8$ a tendência de elementos de baixa qualidade tende a diminuir.

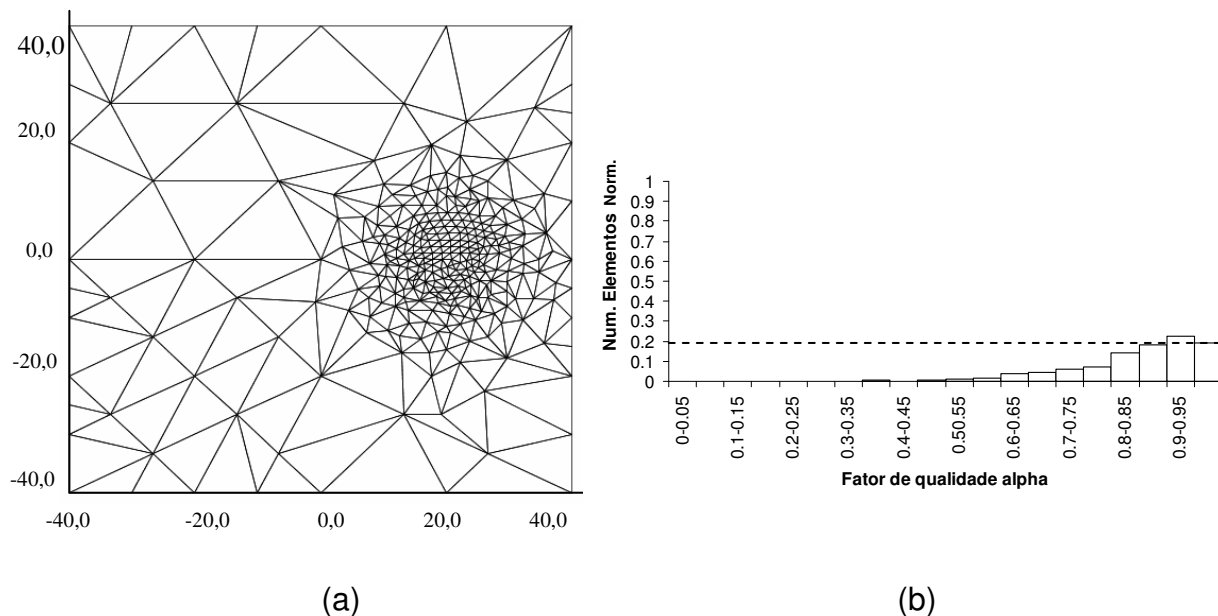


Fig. 5.2: (a) Malha obtida após a técnica de Delaunay, e o seu (b) histograma do fator de qualidade em função do número de elementos normalizados.

Ainda em relação ao histograma da Figura 5.2 (b), o número de elementos na faixa $0,95 \leq \alpha \leq 1,0$, intervalo considerado de critério rigoroso de análise, é aproximadamente 0,19, um valor relativamente baixo. Caso se leve em consideração uma faixa mais ampla de $0,8 \leq \alpha \leq 1,0$, tem-se uma concentração de, aproximadamente, 0,74. Esta quantidade é menor do que as obtidas pelas suavizações laplaciana e a sua variação restrita, que serão vistas a seguir.

Dessa forma, com o objetivo de mudar a distribuição do histograma, aumentando a concentração na faixa de $0,95 \leq \alpha \leq 1,0$, após a suavização laplaciana simples de dez iterações, tem-se o resultado visto na Figura 5.3 (a).

Na Figura 5.3 (b), a primeira instância, nota-se, para o fator de $0,95 \leq \alpha \leq 1,0$, um número de elementos em torno de 0,58, ou seja, três vezes maior que o histograma da Figura 5.2 (b), que é de 0,19. Ao levar em consideração a faixa de $0,8 \leq \alpha \leq 1,0$, temos uma concentração em torno de 0,9. Esses resultados mostram a importância da aplicação da suavização laplaciana para a obtenção de triângulos mais próximos da condição regular.

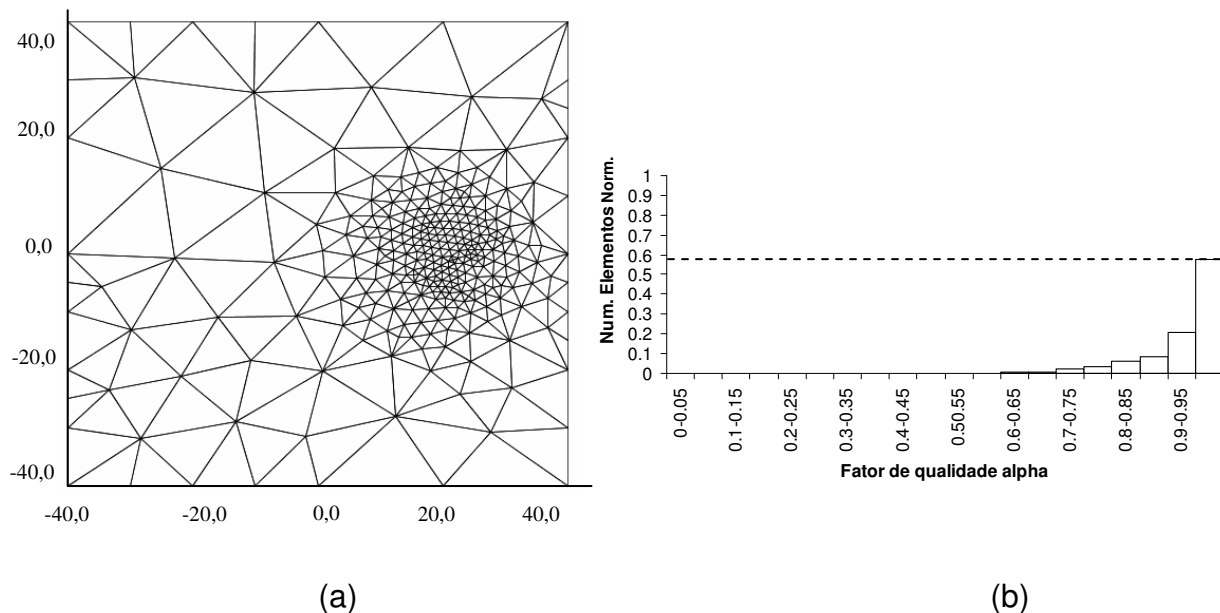


Fig. 5.3: (a) Malha obtida com a aplicação da suavização laplaciana após a técnica de Delaunay, e o seu (b) histograma do fator de qualidade em função do número de elementos normalizados.

Partindo-se, novamente, da malha da Figura 5.2 (a), tem-se a malha da Figura 5.4 (a) e o histograma da Figura 5.4 (b), após a suavização laplaciana restrita, onde a concentração dentro da faixa, $0,8 \leq \alpha \leq 1,0$, se dá também próximo a 0,9. Entretanto, nota-se que a distribuição dos elementos não consta apenas, principalmente, em $0,95 \leq \alpha \leq 1,0$, como observado na Figura 5.3 (b); mas estão distribuídos na faixa $0,9 \leq \alpha \leq 1,0$. Isso se dá devido a esta técnica não modificar quaisquer nós cuja qualidade final da malha comprometa os elementos pertencentes àquele nó central. Nessa técnica, o número de iterações é responsável pela mudança da malha obtida, levando em conta tanto a qualidade inicial como a final da malha. Dessa forma, elementos que possuíam qualidade inicial aceitável, já não sofreram modificações, uma vez que após essa modificação, sua qualidade final será menor do que a obtida inicialmente. Diferentemente da suavização laplaciana, é comum todos os elementos sofrerem modificações, sem a garantia de melhoria na qualidade. Mas, neste caso isolado, leva-se grande parte dos elementos a melhorarem seu aspecto qualitativo. Portanto, se utilizado corretamente, a suavização laplaciana pode levar vantagem em relação a sua variação restrita.

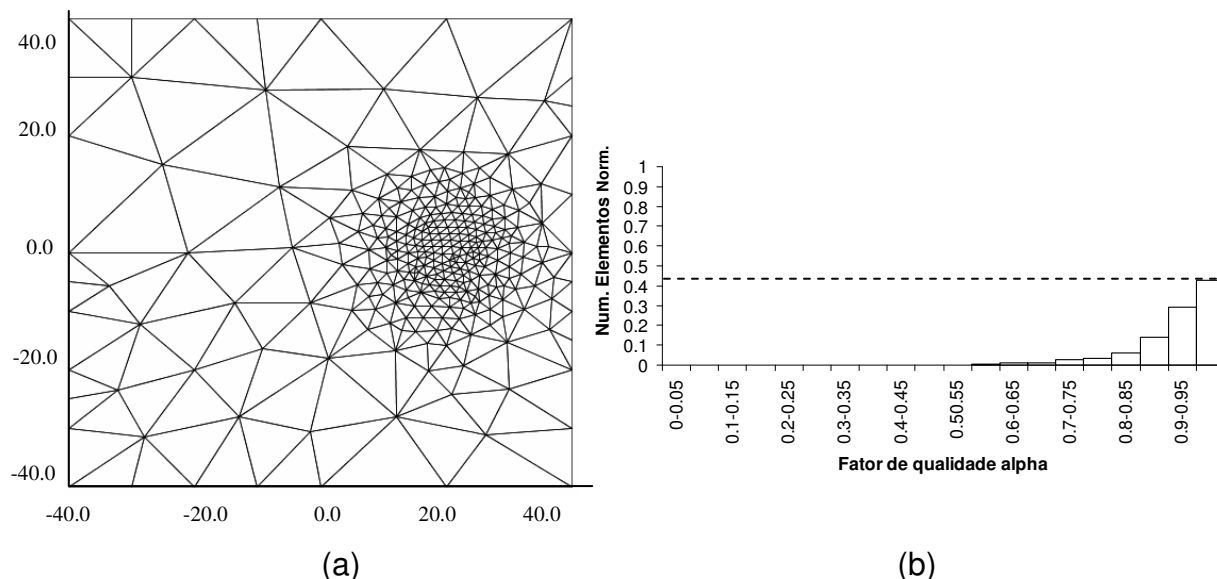


Fig. 5.4: (a) Malha obtida com a aplicação da suavização laplaciana restrita após a técnica de Delaunay, e o seu (b) histograma do fator de qualidade em função do número de elementos normalizados.

5.2 Comparação de Estruturas

É importante expor que, nas próximas seções as dimensões das estruturas não são relevantes, bastando à representação da largura e a altura das mesmas, por variáveis, sem valores definidos. Dessa forma, as dimensões reais, de cada exemplo, foram omitidas.

A aplicação da suavização sobre as malhas geradas, é a laplaciana simples, pela simplicidade na implementação e ter apresentado resultados satisfatórios.

5.2.1 Refinamento de um Guia Retangular

A primeira estrutura representa um guia óptico tipo retangular. Neste exemplo, adiciona-se um canal externo e interno. Define-se neste exemplo, a largura total da estrutura sendo b , e esta é dez vezes maior que a largura do canal interno, $\frac{b}{10}$. Ambas as dimensões são mostradas na Figura 5.5 (a).

Para uma geometria simples de seções retangulares, a malha inicial é gerada automaticamente sem a necessidade de se instaurar uma configuração inicial. Em relação à questão de implementação, o código definido gera a malha inicial mostrada na Figura 5.5 (b).

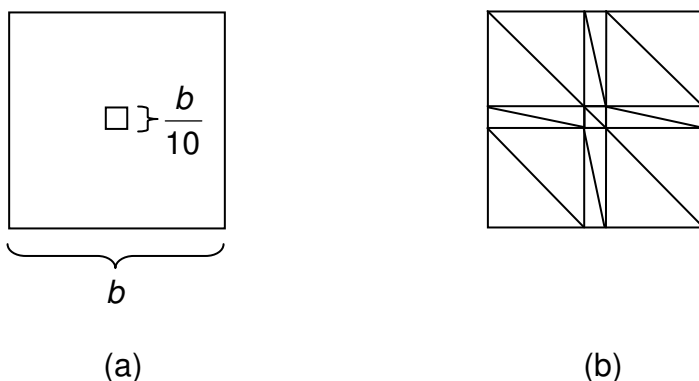


Fig. 5.5: (a) Esboço de um guia óptico retangular e a sua (b) malha inicial.

Para o refinamento, utilizou-se do algoritmo de Delaunay em conjuntura com a técnica baricêntrica, descrita no Capítulo 4. Após este processo, aplicou-se a suavização laplaciana com dez iterações. Esse valor foi suficiente, pois nenhuma mudança significativa da posição dos nós ocorreu, deixando a malha em estado de equilíbrio.

Como parâmetro estipulado, a malha sofreu uma divisão estimada de 3000 elementos lineares (três nós). Como resultado, tem-se que a malha produzida na Figura 5.6 (a) possui 3534 elementos e 1780 nós. O detalhe da região do canal interno pode ser visto na Figura 5.6 (b), com um aumento de seis vezes. Esta última figura mostra claramente a definição da interface, local que apresenta maior densidade de refinamento.

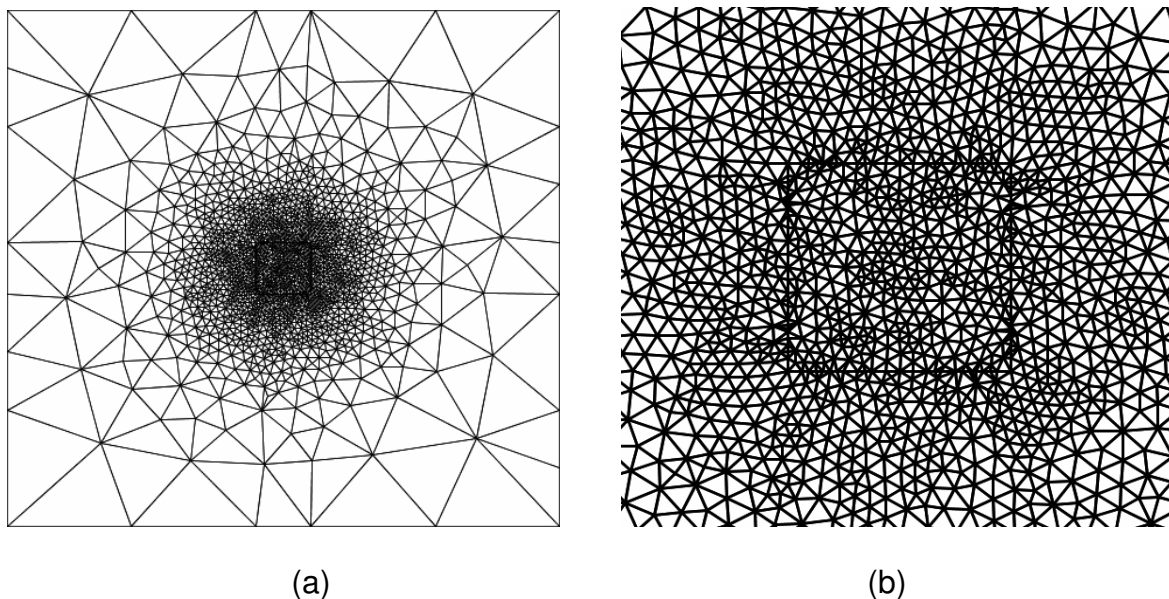


Fig. 5.6: Guia óptico retangular e sua (a) geração de malha numa estrutura completa, e ao lado, o (b) detalhe da região do canal interno, aumentado em seis vezes.

No gráfico da Figura 5.7 (a), mostra-se a qualidade dos elementos em função do número de elementos normalizados. Em torno de 72% dos elementos possui $0,9 \leq \alpha \leq 1,0$ e 15% para $0,8 \leq \alpha < 0,9$. De forma geral, a qualidade desta malha varia de 0,5501 a 0,9996.

Na Figura 5.7 (b), observa-se a distribuição de ângulos internos. O menor ângulo é de 18,23 graus, enquanto o maior ângulo é de 119,59 graus. Este histograma possui 62% dos ângulos internos entre 50 e 70 graus, o que denota a existência de muitos triângulos próximos à condição regular.

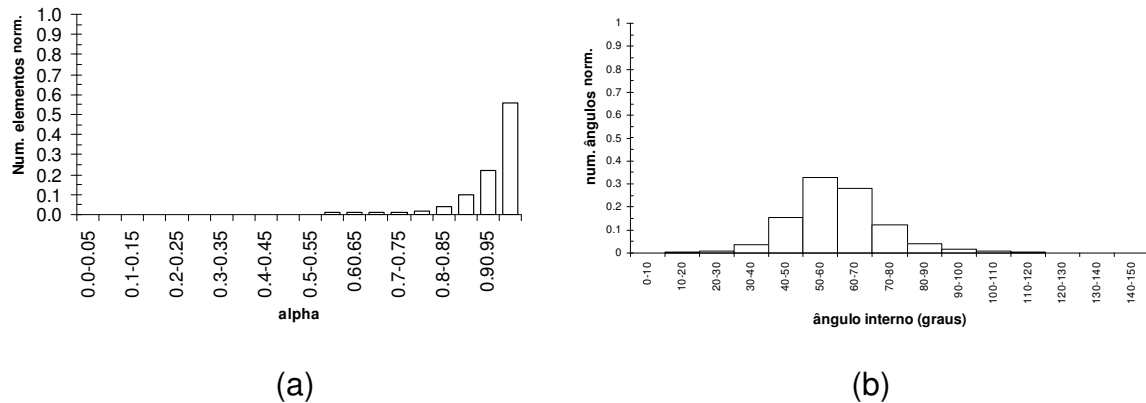


Fig. 5.7: Gráficos de distribuição. (a) histograma de qualidade, (b) histograma de ângulos internos.

Os resultados anteriores são semelhantes aos obtidos em [2]. Entretanto, em [2], o estudo é realizado em um quarto da estrutura, por considerar a simetria do canal. Nesse estudo, a geração de malha em uma estrutura completa mostrou a adaptação dos elementos diante das condições de interface do canal interno e externo. Assim, diferentemente dos trabalhos realizados em [2] e [4], neste estudo, realizou-se o refinamento na estrutura completa, portanto, um quarto da malha gerada não é necessariamente simétrico em relação aos outros quadrantes.

Os resultados obtidos através de uma malha de 3012 elementos e 1551 nós mostram aproximadamente 90% dos elementos com α acima de 0,9. Além disso, cerca de 65% dos ângulos internos estão próximos a 60 graus. Os resultados desses elementos, de acordo com [2] e [4], são considerados de excelente qualidade. Ainda num trabalho semelhante, com a mesma estrutura desenvolvida por [5], aproximadamente 85% dos elementos têm qualidade superior a 0,9 e a distribuição dos ângulos internos mostra que 60% destes estão entre 50 e 70 graus.

Em relação à obtenção de malhas para a mesma estrutura, tem-se o gráfico da Figura 5.8, que mostra o fator de qualidade α em função do número de elementos normalizados. Observa-se uma comparação entre diversas malhas geradas referente à estrutura da Figura 5.5 (a). Os resultados obtidos desde uma malha menos refinada com 1086 até a mais densa com 10664 elementos, mostram que conforme o número de elementos aumenta, a proporção da concentração de elementos na faixa de $0,9 \leq \alpha \leq 1,0$ tende a aumentar. Por exemplo, para 6176 elementos, aproximadamente 83% dos elementos estão dentro dessa faixa e para 10664 elementos esse número é de 86%. Esse crescimento pode ser acompanhado na Tabela 5.1.

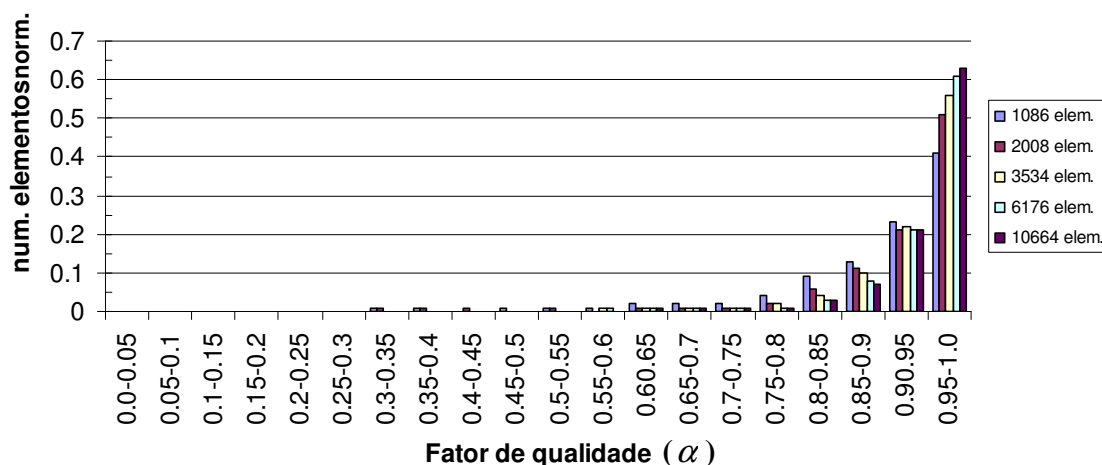


Fig. 5.8: Gráficos de distribuição do número de elementos normalizados em função do fator e qualidade alpha para cinco densidades de refinamento.

Número de elementos	% de elementos na faixa $0,9 \leq \alpha \leq 1,0$
1086	65
2008	73
3531	79
6176	83
10664	86

Tab. 5.1: Tabela de crescimento da porcentagem de elementos na faixa de $0,9 \leq \alpha \leq 1,0$.

A Figura 5.9 indica o tempo de execução, em segundos, em função do número de elementos produzidos. Pode-se observar claramente que à medida que a estrutura se torna mais refinada, maior o tempo computacional. Esse aumento está condizente com os trabalhos vistos em [2] e [5].

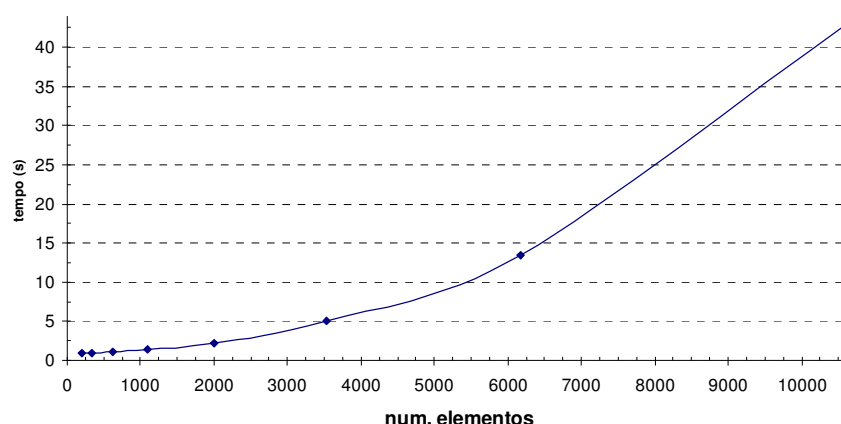


Fig. 5.9: Gráfico do tempo de execução.

Ressalta-se que o gráfico de tempo de execução deve ser comparado com trabalhos da literatura de [2] e [5], pois a conjuntura das técnicas baricêntrica, bissecção, Delaunay e a suavização laplaciana encontram-se fundamentadas nesses estudos. Assim, não se pode realizar uma comparação de tempo de processamento com programas comerciais como GiD [35], onde as técnicas implementadas, o código-fonte e a documentação teórica do programa não são acessíveis aos pesquisadores.

5.2.2 Refinamento de um Guia Tipo Microfita

Nesta seção apresentam-se alguns exemplos de malhas obtidas no refinamento de um guia tipo microfita. Essa estrutura é interessante devido a sua cavidade interna. Devido à presença dessa fina região condutora, ocorre uma grande variação no aspecto dos elementos. Este guia é composto de uma fina

região condutora localizada na parte central e colocada sobre um substrato dielétrico. O material da camada superior é o ar.

O esboço dessa estrutura pode ser visto na Figura 5.10. A estrutura total possui largura e altura, respectivamente, a e $\frac{a}{2}$. A altura do dielétrico é $\frac{a}{4}$. A região condutora de largura $b = \gamma.a$ e altura $c = \kappa.a$. Essas características foram utilizadas por [2].

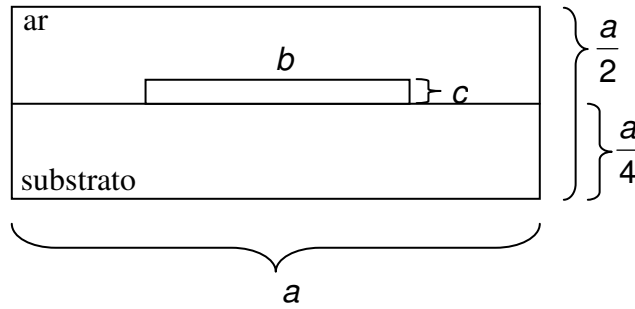


Fig. 5.10: Estrutura de um guia tipo microfita.

A geração de malhas numa estrutura de cavidade interna pode ser realizada da seguinte forma. Considere um conjunto finito de pontos $V = \{v_0, v_1, v_2, \dots, v_n\}$, $v_i \in \mathbb{R}^2, 0 \leq i \leq n$, sabe-se que cavidade interna é delimitada pelos seguintes valores:

$$\begin{cases} X_{\min} \leq x_i \leq X_{\max} \\ e \\ Y_{\min} \leq y_i \leq Y_{\max} \end{cases}$$

onde (x_i, y_i) são as coordenadas do vértice v_i , e (x_{\min}, y_{\min}) e (x_{\max}, y_{\max}) são, respectivamente, o vértice inferior esquerdo e o vértice superior direito da cavidade.

Na Figura 5.11 observa-se a geração de malha sem as condições delimitadas. Na Figura 5.12, todos os elementos que possuem pontos dentro da região delimitada são removidos.

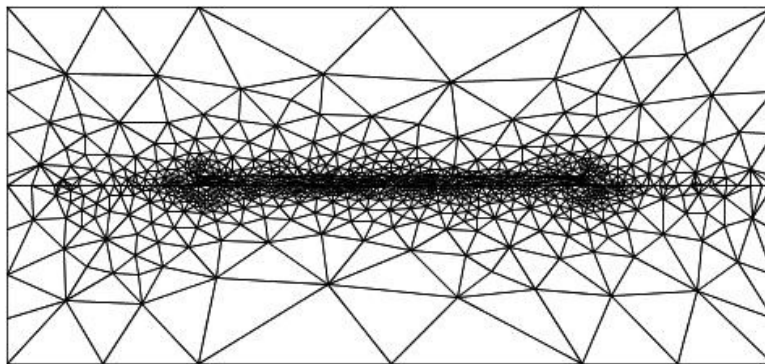


Fig. 5.11: Geração de malha sem a cavidade delimitada.

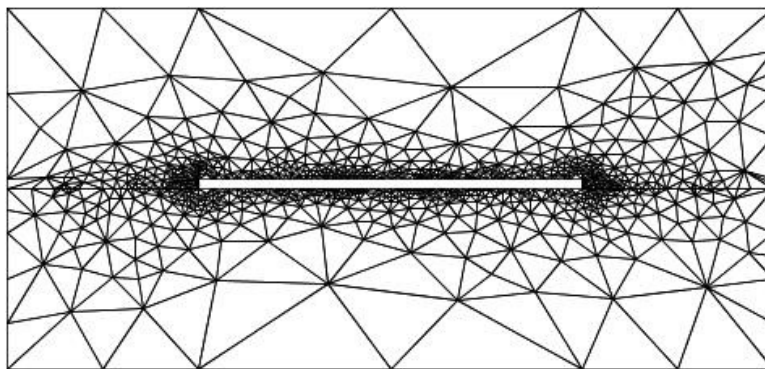


Fig. 5.12: Geração de malha com a cavidade delimitada.

Conforme mostrado na Figura 5.12, ainda é possível melhorar a distribuição dos elementos nas regiões próximas ao limite da interface externa da estrutura. Para isso, o número de vértices concentrados entre a divisão (em relação ao eixo x) delimitada pelos extremos da cavidade interna, nas interfaces, deve sofrer um incremento. A divisão central pode ser vista na Figura 5.13, nas interfaces de cor vermelha.

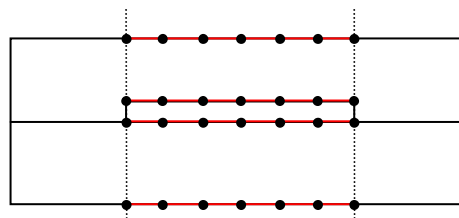


Fig. 5.13: Adição de vértices na divisão central.

O aumento da distribuição de pontos sobre a interface produz a malha, que pode ser vista na Figura 5.14 (a), e a sua ampliação em três vezes, na Figura 5.14 (b). A opção de alteração do número de vértices no contorno mostra a importância de adaptação dos algoritmos em geral, frente às condições indeterminadas no refinamento de uma estrutura.

Nas Figuras 5.14 (a) e (b) observa-se a concentração acentuada de elementos nos cantos e uma leve distribuição deles ao longo da camada. Esse resultado pode ser obtido por meio da soma de várias curvas ou funções, tomando como exemplo as gaussianas.

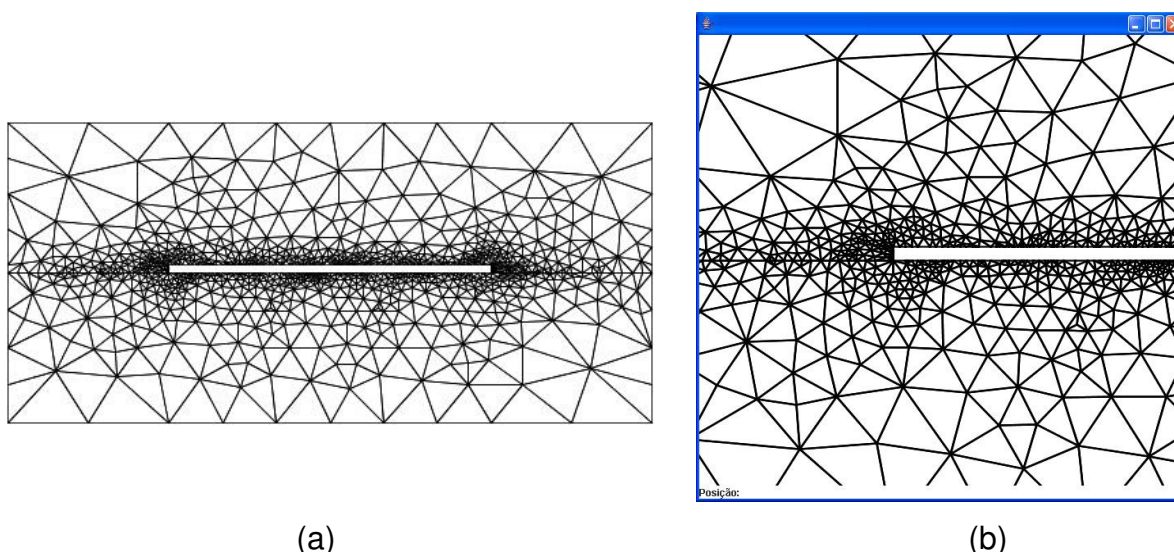


Fig. 5.14: Adaptação da (a) malha através da adição de novos vértices na interface, e a sua (b) ampliação em três vezes.

A fim de verificar os aspectos qualitativos, a Figura 5.15 (a) mostra o número de elementos em função do fator de qualidade. Tem-se que cerca de 87% dos elementos possui qualidade superior a 0,8, o que é considerado um resultado próximo aos resultados obtidos por [2].

O histograma da Figura 5.15 (b) mostra o número de elementos em função dos ângulos internos. Tem-se que 40% dos ângulos internos concentram-se entre 50 e 70 graus, com baixa quantidade de ângulos inferiores a 20 graus e superiores a 100 graus.

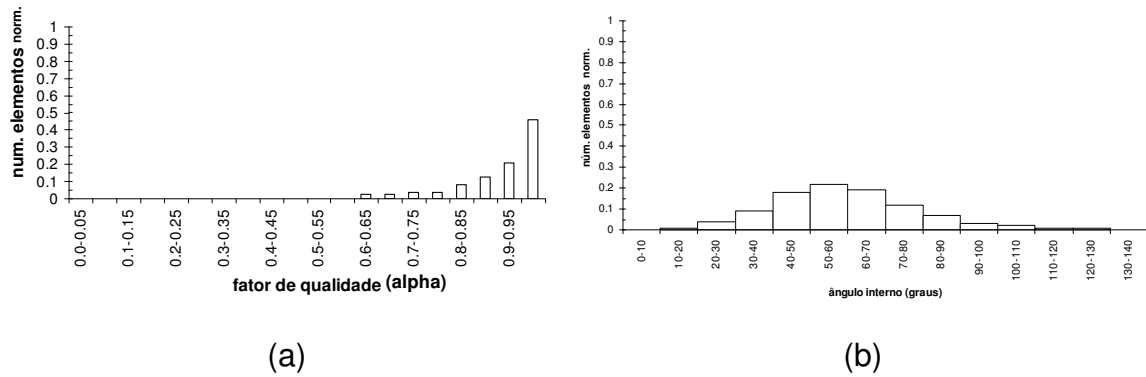


Fig. 5.15: Distribuição do (a) fator de qualidade e dos (b) ângulos internos em função do número de elementos normalizados.

De acordo com a literatura, embora os resultados sejam satisfatórios, observa-se a presença de ângulos agudos. Esse problema pode ser contornado se introduzirmos outras técnicas de aperfeiçoamento de malha no algoritmo, no entanto, presume-se que o tempo de processamento aumentará. Também se podem aplicar testes com outras técnicas de refinamento como o algoritmo de Bowyer/Watson e avanço de fronteira.

5.2.3 Refinamento de um Guia Retangular Parcialmente Preenchido

Uma estrutura vista em [5] é um guia retangular parcialmente preenchido por um dielétrico cuja dimensão pode ser vista na Figura 5.16. A estrutura total possui largura $2a$ e altura a . Apenas por questões representativas, as dimensões reais, dito novamente, serão omitidas focando-se apenas no estudo da adaptação dos elementos.

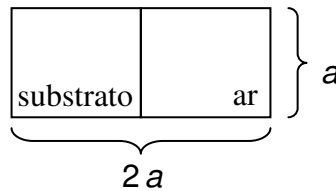


Fig. 5.16: Esboço de estrutura de guia retangular parcial.

Como se pode notar, a região crítica para esse domínio é nas proximidades da interface entre os dois meios. Com relação à geração de malha nesta configuração, temos que na Figura 5.17 (a) produziu-se 3198 elementos e 1632 nós. Pode-se também gerar uma malha mais refinada nos cantos da estrutura. Assim, como no exemplo visto na Figura 5.14 (a), na adição de nós de contorno em relação ao eixo x, neste caso pode-se realizar a inserção de nós nos cantos da estrutura pelo eixo y.

O resultado, observado na Figura 5.17 (b), mostra a produção de 4040 elementos e 2049 nós. Entretanto, realiza-se uma comparação entre os histogramas de qualidade de cada uma das malhas obtidas.

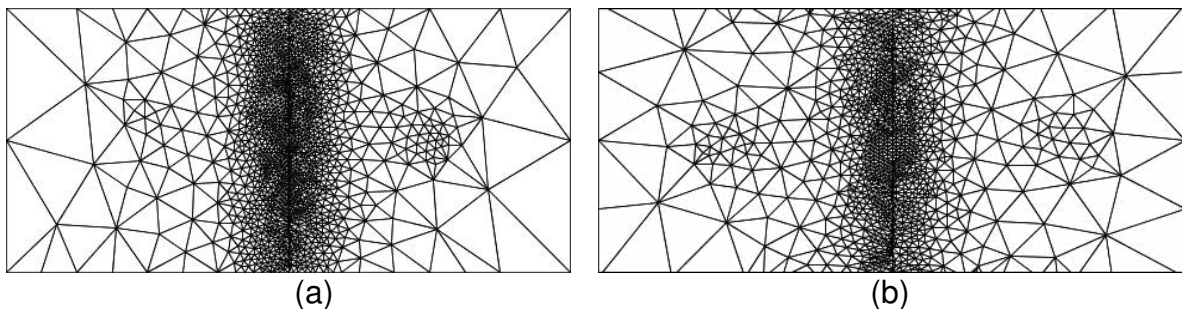


Fig. 5.17: Refinamento de guia retangular parcial. (a) Refinamento produzido (b) Refinamento modificado.

Os histogramas vistos na Figura 5.18 (a) e (b) associam-se à geração de malha da Figura 5.17 (a), enquanto que as Figuras 5.19 (a) e (b) associam-se à da Figura 5.17 (b).

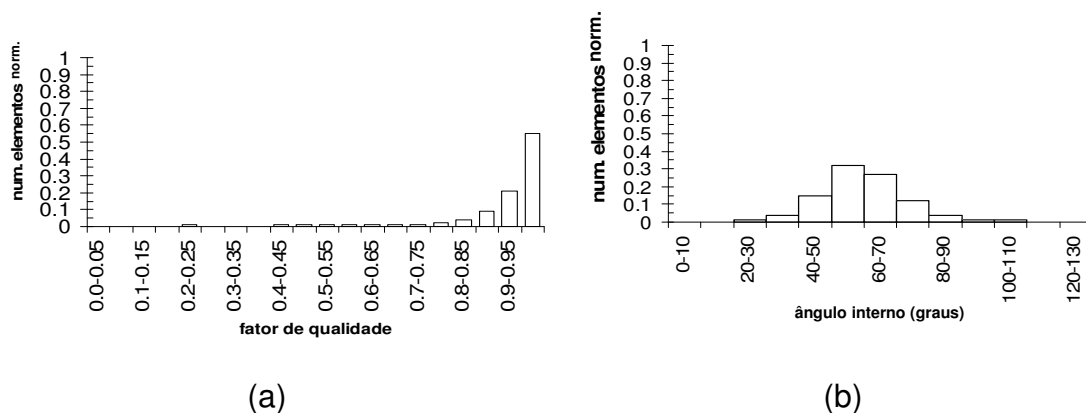


Fig. 5.18: Distribuição do (a) fator de qualidade e dos (b) ângulos internos em função do número de elementos normalizados, da Figura 5.17 (a).

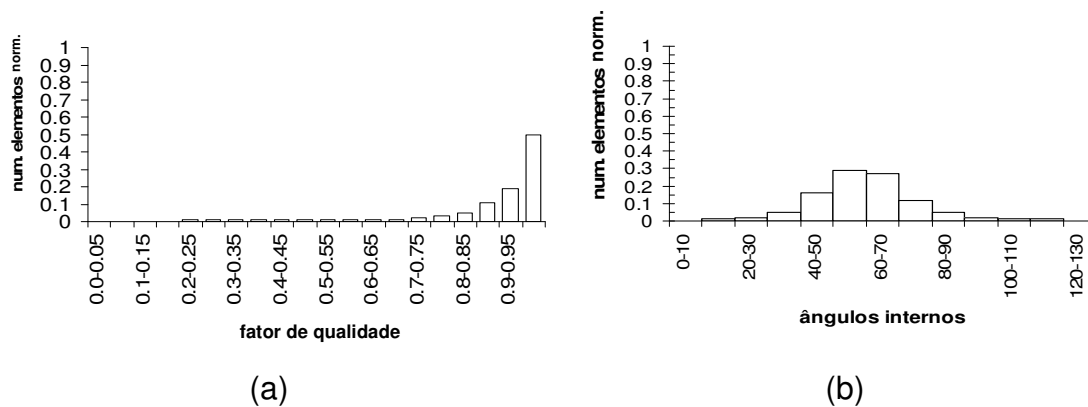


Fig. 5.19: Distribuição do (a) fator de qualidade e dos (b) ângulos internos em função do número de elementos normalizados, da Figura 5.17 (b).

Ambos os histogramas de qualidade possuem distribuição de valores semelhantes. A figura 5.18 (a) 71%, e em 5.19 (a) 70% dos elementos correspondem ao fator 0,9. Indiferente, a distribuição de ângulos internos também mantém semelhanças, no primeiro 59% e 57% no segundo. Dessa forma, foi possível alterar a distribuição dos elementos, mantendo inalterado o aspecto qualitativo da malha.

Conclui-se que os resultados apresentados nesta seção, como em todas as anteriores, exibem coerência de valores qualitativos em relação aos descritos na literatura. Além disso, a adaptação da malha mostra que o sistema traçado na Seção 5.1, ou seja, a descrição dos algoritmos desenvolvidos mostra a relação de confiança e eficiência no que diz respeito ao refinamento desejado.

Como foi dito na introdução deste capítulo, a alteração do número de nós de contorno mostra a flexibilidade da geração de malhas, possibilitando realizar uma comparação entre malhas distintas das mesmas estruturas, além de permitir realizar futuros trabalhos relacionados à geração de malhas, partindo-se da definição do tamanho das arestas dos elementos pertencentes às interfaces.

Capítulo 6

Testes e Simulações

Neste capítulo, realizou-se a avaliação das malhas através da análise dos resultados numéricos. Para isso, as estruturas utilizadas foram submetidas a um refinamento, que produziram seus respectivos arquivos de saída. Esses arquivos contêm informações que servem como base de entrada para as simulações realizadas mediante o programa Aníbal, que calcula o índice efetivo modal dos guias ópticos. Os índices foram obtidos através das malhas produzidas pelo gerador deste projeto, e também pelo programa GiD. Assim, mantendo-se referência aos valores obtidos no GiD, obteve-se os gráficos do $\log(\text{erro})$ em função de λ , e da queda linear do $\log(\text{erro})$ em função do número de elementos.

Diferentemente do Capítulo 5, utilizou-se valores de dimensões reais das estruturas. E por questão de organização, este capítulo divide-se em três seções, cada uma referindo-se a uma estrutura distinta. Um breve estudo sobre as aplicações dos guias ópticos pode ser obtido nas notas de leitura de Pissolato Filho [41]

6.1 Recursos Utilizados

Nesta seção dá-se o início a uma descrição dos recursos e programas utilizados no desenvolvimento do trabalho. Quanto aos recursos de *hardware*

utilizados, têm-se o processador Intel Pentium M 1.73 GHz; memória RAM 1 GB. O sistema operacional é o *Microsoft Windows XP*.

Os ambientes de trabalho para desenvolvimento e testes dos algoritmos foram: Compilador/Interpretador Eclipse SDK v. 3.2.2 [42] e Netbeans 4.1 [43], ambos utilizados para a programação em linguagem Java. Ressalta-se que esta linguagem foi escolhida de acordo com a preferência e afinidade do autor. Entretanto, não se descarta o uso de outras linguagens, como o Fortran orientado à objetos e C++. No Apêndice C.3, descreve-se sucintamente outra possibilidade, como a integração da linguagem Java ou C++ com OpenGL. As linguagens C e Fortran procedural não serão abordadas, devido ao código estruturado. Informações sobre o uso do Fortran procedural pode ser obtido em [2].

As ferramentas auxiliares utilizadas para a obtenção dos resultados, além dos algoritmos de refinamento produzidos, foram:

- Programa de pré-processamento comercial, denominado GID v. 7.2 [35];
- Um programa de geração de malhas utilizando o refinamento de bissecção simples, denominado HMesh, desenvolvido pelo Prof. Dr. Hugo E. H. Figueroa;
- Um programa que realiza cálculo de índice efetivo modal (modo fundamental) de guias ópticos, denominado ANIBAL, desenvolvido pelo Prof. Dr. Hugo E. H. Figueroa;

6.2 Refinamento de um Guia Óptico Retangular

No primeiro exemplo, realiza-se uma análise da constante de propagação em uma geometria de um guia retangular óptico. Na Figura 6.1 (a) são introduzidas paredes perfeitamente casadas (PML - *Perfectly Matched Layers*) de espessura $2\ \mu\text{m}$. A estrutura em estudo é um guia retangular de dimensão interna $4\ \mu\text{m} \times 4\ \mu\text{m}$ e o limite computacional têm dimensão $16\ \mu\text{m} \times 16\ \mu\text{m}$.

A Figura 6.1 (b) mostra as coordenadas da geometria definida num plano, cuja estrutura interna do guia óptico varia $x = [-2, 2] \mu m$ e $y = [-2, 2] \mu m$, e o início da margem superior esquerda e o início da margem inferior direita da PML são, respectivamente, $(-6, 6) \mu m$ e $(6, -6) \mu m$. Informações sobre a saída de arquivos, com a definição das condições de contorno podem ser obtidas no Apêndice D.

Na Figura 6.1 (c), dois tipos de materiais foram utilizados. O índice de refração da casca é 1,00 e o índice de refração do núcleo é 1,45. Os comprimentos de onda utilizados foram: $\lambda = 1,3 \mu m, 1,35 \mu m, 1,4 \mu m, 1,45 \mu m, 1,5 \mu m, 1,55 \mu m, 1,6 \mu m, 1,65 \mu m$ e $1,7 \mu m$.

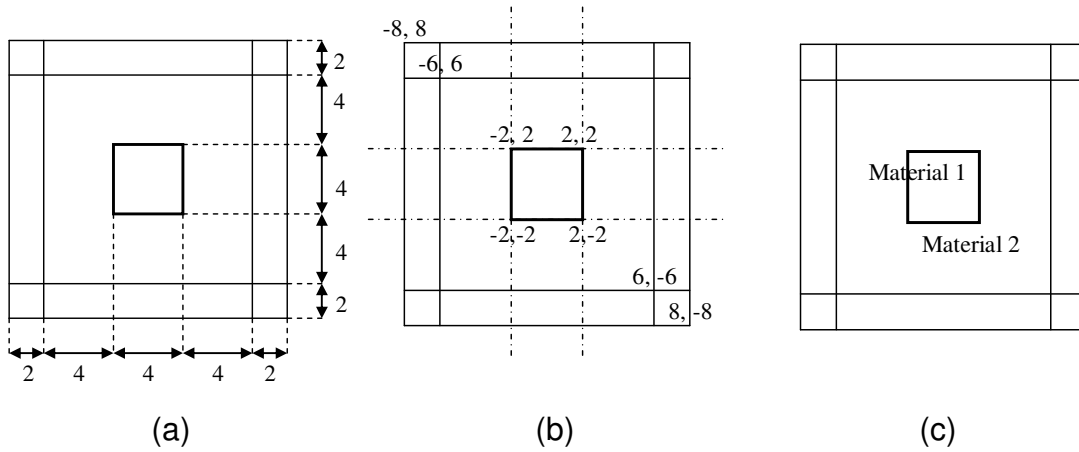


Fig. 6.1: Definindo a (a) proporção do guia óptico retangular, as (b) coordenadas desejadas, e os (c) tipos de materiais.

Realizaram-se cinco testes, porém com propriedades de densidade de refinamento distintas, ou seja, cada teste contou com uma variação da excitação do feixe gaussiano, registrados com os seguintes nomes: geom1_dens1, geom1_dens2, geom1_dens3, geom1_dens4 e geom1_dens5. A Tabela 6.1 mostra uma relação desses nomes em função do número ascendente de elementos estimados. Nessa tabela são apresentados também o número total de elementos e os nós gerados. O raio de concentração é definido na região central da estrutura, e é determinado pelos seguintes parâmetros: A , a amplitude; (x_0, y_0) , o centro; e (d_1, d_2) , os raios da função gaussiana. Quanto maior a concentração de elementos, maior será o raio do feixe ou *spot*, que será vista nas figuras a seguir.

características saída	Num. elem. estimados	Num. elementos	Num. nós	Parâmetros da função gaussiana		
				A	$(x_0; y_0)$	$(d_1; d_2)$
geom1_dens1	700	750	401	1,0	(0,0; 0,0)	(0,2; 0,2)
geom1_dens2	1900	1916	984	1,0	(0,0; 0,0)	(2,0; 2,0)
geom1_dens3	2300	2320	1186	1,0	(0,0; 0,0)	(2,5; 2,5)
geom1_dens4	3600	3632	1842	1,0	(0,0; 0,0)	(2,5; 2,5)
geom1_dens5	7100	7140	3596	1,0	(0,0; 0,0)	(2,5; 2,5)

Tab. 6.1: Características do refinamento de um guia óptico retangular.

As Figuras 6.2 (a) – (e), mostram a geração de malhas, respectivamente de acordo com a ordem da Tabela 6.1, ou seja do menor ao maior número de elementos. Nestas figuras, o raio de concentração de elementos tende a aumentar.

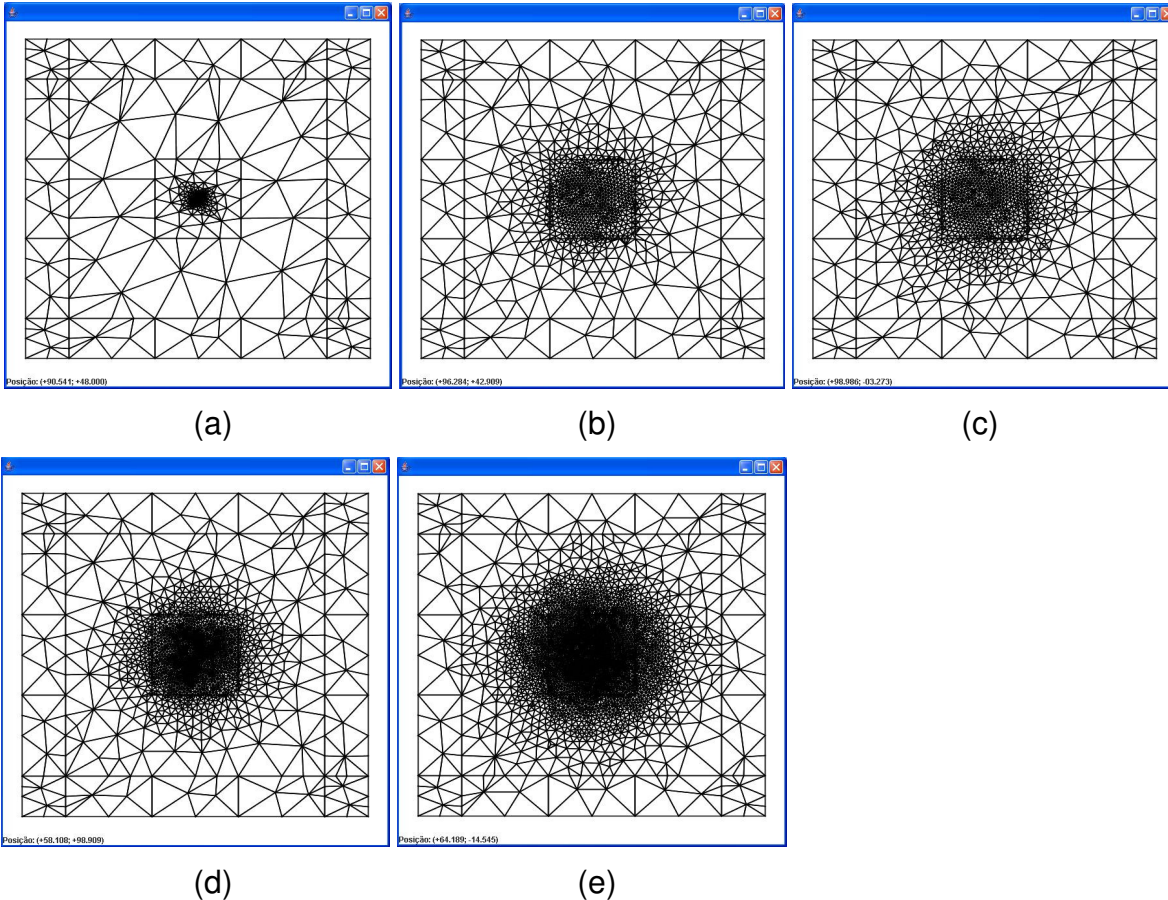


Fig. 6.2: Geração de malhas de um guia óptico retangular e as suas seguintes identificações de saída: (a) geom1_dens1, (b) geom1_dens2, (c) geom1_dens3, (d) geom1_dens4, e (e) geom1_dens5.

A Tabela 6.2 mostra a relação de λ e o índice efetivo modal calculado para cada refinamento obtido das Figuras 6.2 (a) – (e). Para o cálculo do índice efetivo, utilizou-se o programa denominado ANIBAL. Na Tabela 6.2, Ge_1 indica o resultado numérico através do refinamento obtido pelo programa GiD, e os demais resultados dens1, dens2, dens3, dens4, dens5 foram produzidos no gerador de malha deste trabalho. Em Ge_1 produziu-se 17864 elementos e 36085 nós, com simulação de elementos de ordem quadrática, enquanto a obtenção dos demais resultados utilizou-se a ordem linear.

lambda	Ge₁	densidade₁	densidade₂	densidade₃	densidade₄	densidade₅
1,3	1,433963	1,431458	1,433764	1,433808	1,433884	1,433952
1,35	1,432796	1,430073	1,432575	1,432623	1,432704	1,432777
1,4	1,431592	1,428645	1,431350	1,431401	1,431488	1,431566
1,45	1,430353	1,427174	1,430089	1,430144	1,430236	1,430320
1,5	1,429080	1,425662	1,428793	1,428852	1,428949	1,429038
1,55	1,427773	1,424109	1,427463	1,427525	1,427628	1,427723
1,6	1,426433	1,422516	1,426098	1,426164	1,426273	1,426373
1,65	1,425061	1,420883	1,424699	1,424770	1,424884	1,424990
1,7	1,423657	1,419211	1,423268	1,423343	1,423463	1,423575

Tab. 6.2: Comprimento de onda e o cálculo do índice modal, do guia retangular.

A partir dos resultados numéricos obtidos na Tabela 6.2, realizou-se o cálculo dos $\log(\text{erro})$, através de (6.1). Os resultados podem ser vistos na Tabela 6.3, onde foram mantidos como referência, os resultados Ge_1 produzidos pelo GiD.

$$\text{Log}(\text{erro}_x) = \text{Log} \left(\frac{\text{dens}_x - Ge_1}{Ge_1} \right) \quad (6.1)$$

Log(ERRO1)	Log(ERRO2)	Log(ERRO3)	Log(ERRO4)	Log(ERRO5)
-2,757666157	-3,858256193	-3,966777518	-4,259886631	-5,135824578
-2,7211516	-3,813264602	-3,918496332	-4,193788865	-4,892185432
-2,686383692	-3,772430387	-3,875259797	-4,137301896	-4,743048701
-2,653137009	-3,73411686	-3,834964012	-4,085989183	-4,629496256
-2,621264149	-3,697592388	-3,796686184	-4,037996413	-4,534451923
-2,590650995	-3,662450107	-3,759923555	-3,992363083	-4,450765827
-2,561202381	-3,628432931	-3,724365176	-3,948526039	-4,374818927
-2,532835902	-3,595364301	-3,689802097	-3,906127304	-4,30454756
-2,505478667	-3,563115784	-3,65608562	-3,86492612	-4,238672853

Tab. 6.3: Cálculo do $\log(\text{erro})$, do guia retangular, em referência ao GiD.

A Figura 6.3 (a) mostra o comprimento de onda em função de $\log(\text{erro})$ para as cinco densidades obtidas nas Figuras 6.2 (a) - (e). O gráfico mostra que para cada uma das cinco curvas, quanto maior o comprimento de onda, maior será o $\log(\text{erro})$. A geometria de maior refinamento, dada por Dens5, corresponde à curva mais baixa em relação às demais curvas, já que possui maior densidade de refinamento. Dessa forma, a ordem das curvas deve estar de cima para baixo, respectivamente, do menor ao maior refinamento. O gráfico da Figura 6.3 (b) mostra o número de elementos produzidos em função do $\log(\text{erro})$. Esse resultado foi obtido comparando-se os comprimentos de onda $1,3 \mu\text{m}$, $1,35 \mu\text{m}$, $1,4 \mu\text{m}$, $1,45 \mu\text{m}$, $1,5 \mu\text{m}$, $1,55 \mu\text{m}$, $1,6 \mu\text{m}$, $1,65 \mu\text{m}$, $1,7 \mu\text{m}$. Observa-se uma queda linear do $\log(\text{erro})$ à medida que o número de elementos produzidos cresce. Essa queda é justificada por meio da concentração de elementos localizados na região central do guia retangular óptico, ou seja, à medida que o número de elementos aumenta, o $\log(\text{erro})$ tende a diminuir.

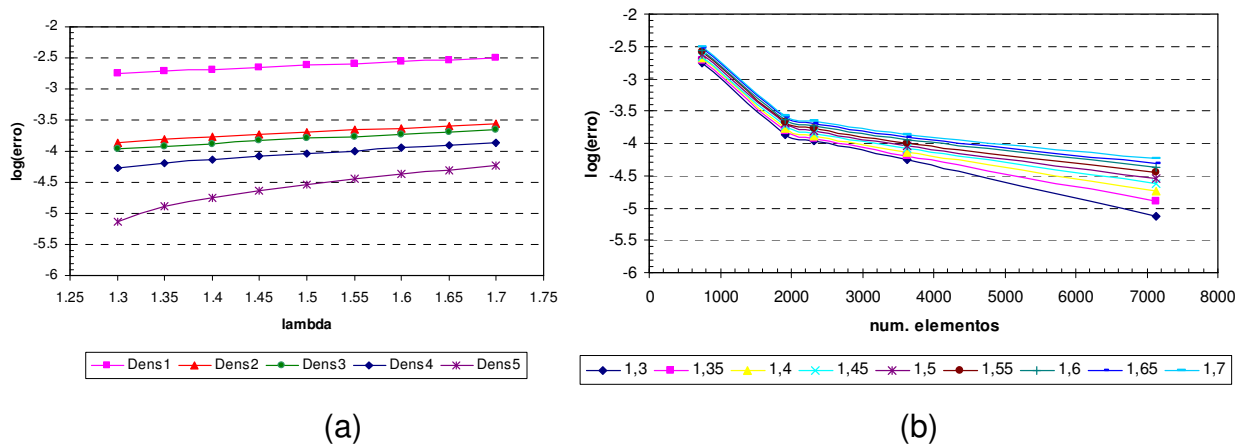


Fig. 6.3: Gráficos do (a) $\log(\text{erro})$ em função de λ , e da (b) queda linear do $\log(\text{erro})$ em função do número de elementos, do guia retangular.

Na próxima seção, será realizado o cálculo numérico através do refinamento de um guia tipo costela. Isso permitirá comparar a geração dos gráficos do $\log(\text{erro})$ em função de λ e a queda linear de $\log(\text{erro})$ em

função do número de elementos, dessa estrutura, com os gráficos obtidos nesta seção.

6.3 Refinamento de um Guia Tipo Costela ('Rib')

No segundo teste, é feita uma análise da constante de propagação em uma geometria de um guia tipo costela. Na Figura 6.4 (a) são introduzidas paredes perfeitamente casadas (PML-*Perfectly Matched Layers*) de espessura $2\mu m$. A estrutura interna em estudo é uma guia com base horizontal inferior de dimensão $4\mu m$, e base superior com $2\mu m$ de dimensão. Os lados verticais da estrutura possuem $2\mu m$ de extensão, a partir dos extremos da base inferior, e outros $2\mu m$ de extensão a partir da base superior. O limite computacional possui dimensão $16\mu m \times 16\mu m$.

A Figura 6.4 (b) mostra as coordenadas da geometria, definidas num plano, cuja estrutura interna do guia óptico é dada por:

$$x = \begin{cases} [-2, +2] \text{ e } y = -1 \text{ ou } y = -2 \\ [-2, -1] \text{ e } y = 0 \\ [+1, +2] \text{ e } y = 0 \\ [-1, +1] \text{ e } y = 2 \end{cases}, \text{ e } y = \begin{cases} [-2, 0], x = \pm 2 \\ [0, +2], x = \pm 1 \end{cases} \quad (6.1)$$

O início da margem superior esquerda e o início da margem inferior direita da PML são, respectivamente, $(-6,6)$ e $(6,-6)$.

Na Figura 6.4 (c), três tipos de materiais foram utilizados. A estrutura interna está dividida em material 1, material 2 e material 3, respectivamente, com índices de refração de $n_1=1,5$, $n_2=1,45$ e $n_3=1,00$. Os comprimentos de onda utilizados foram: $\lambda = 1,3\mu m$, $1,35\mu m$, $1,4\mu m$, $1,45\mu m$, $1,5\mu m$, $1,55\mu m$, $1,6\mu m$, $1,65\mu m$ e $1,7\mu m$.

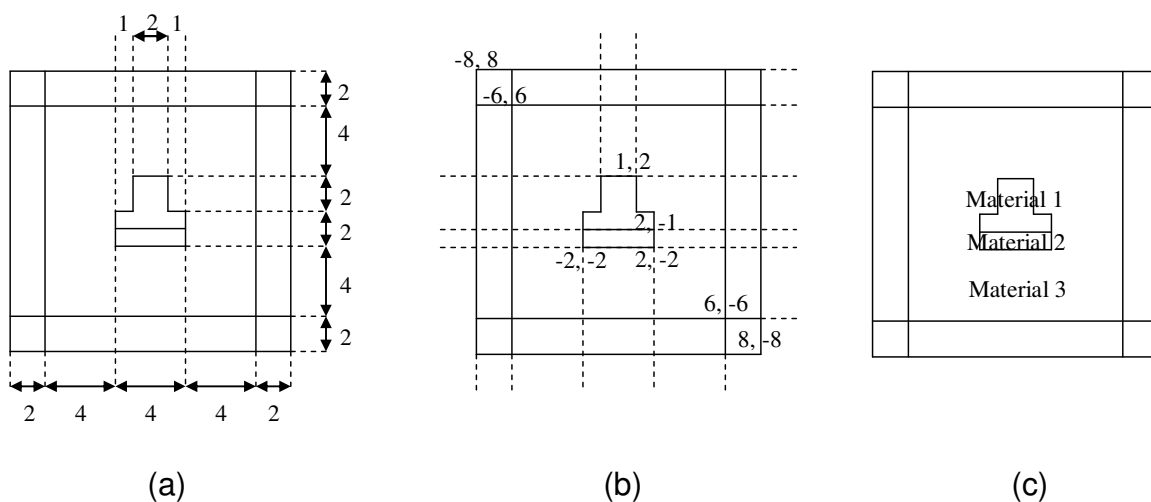


Fig. 6.4: Definindo a (a) proporção do guia tipo costela, as (b) coordenadas desejadas, e os (c) tipos de materiais.

Da mesma maneira que na seção anterior, realizaram-se cinco testes, utilizando-se a geometria da Figura 6.4, que está associada a um número de elementos estimados. Essas informações encontram-se na Tabela 6.4, que mostra a quantidade total de elementos e os nós gerados, além dos parâmetros utilizados na definição dos feixes, dados por A , a amplitude; (x_0, y_0) , o centro; e (d_1, d_2) , os raios da função gaussiana.

características saída	Num. elem. estimados	Num. elementos	Num. nós	Parâmetros da função gaussiana		
				A	(x_0, y_0)	(d_1, d_2)
geom2_dens1	1800	1856	962	1,0	(0,0; 0,0)	(2,0; 2,0)
geom2_dens2	3200	3294	1681	1,0	(0,0; 0,0)	(2,0; 2,0)
geom2_dens3	4400	4458	2263	1,0	(0,0; 0,0)	(1,5; 1,5)
geom2_dens4	5800	5820	2944	1,0	(0,0; 0,0)	(2,5; 2,5)
geom2_dens5	10400	10494	5281	1,0	(0,0; 0,0)	(2,5; 2,5)

Tab. 6.4: Características do refinamento de um guia tipo costela.

As Figuras 6.5 (a) – (e), mostram a geração de malhas, respectivamente, de acordo com a ordem da Tabela 6.1, ou seja, do menor ao maior número de

elementos. Nestas figuras, o raio de concentração de elementos tende a aumentar.

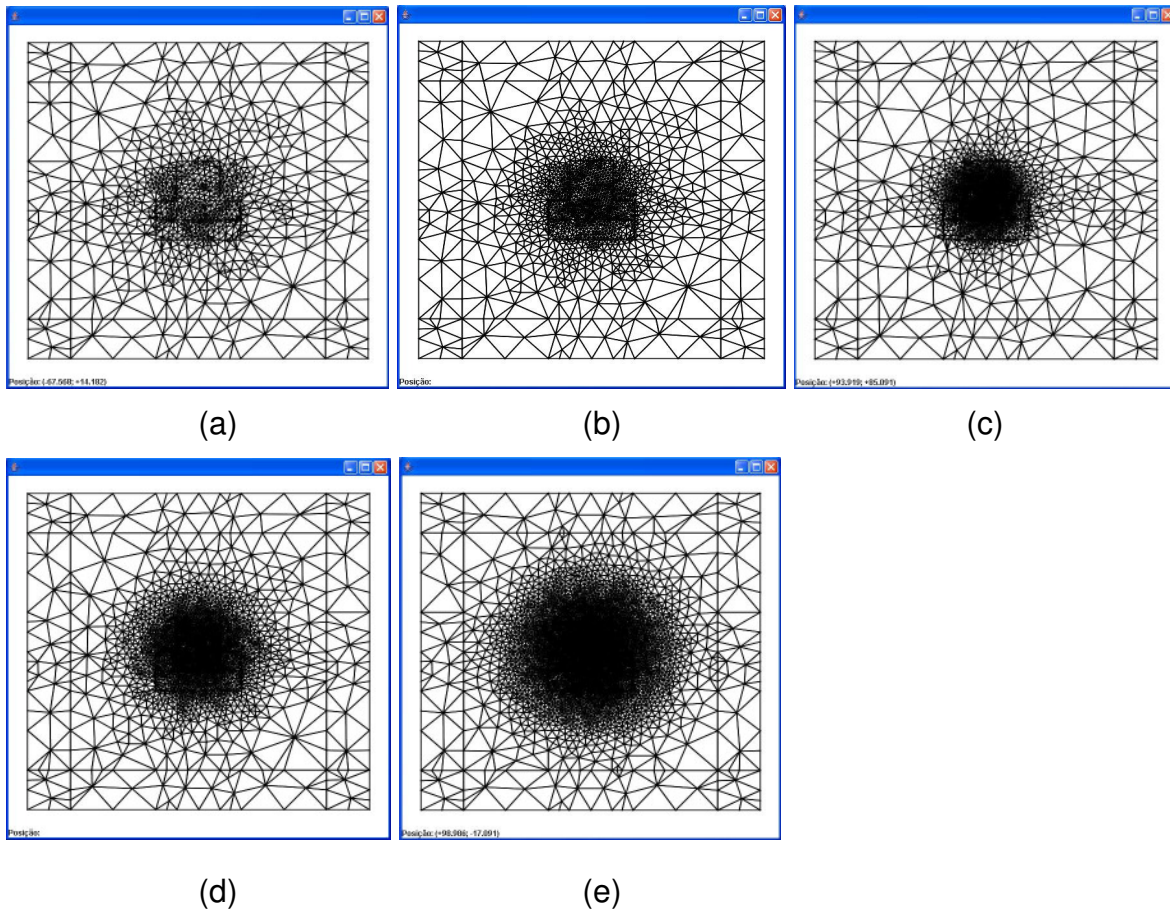


Fig. 6.5: Geração de malhas de um guia tipo costela e as suas seguintes identificações de saída: (a) geom2_dens1, (b) geom2_dens2, (c) geom2_dens3, (d) geom2_dens4, e (e) geom2_dens5.

Na Tabela 6.5, em Ge1 produziu-se 17900 elementos e 36157 nós, com simulação de elementos de ordem quadrática, enquanto nos demais resultados utilizaram-se ordem linear. De acordo com as Tabelas 6.5 e 6.6, lembrando que o cálculo do $\log(\text{erro})$ foi mostrado em (6.1), obteve-se os gráficos mostrados na Figura 6.6 (a) e (b). Na Figura 6.6 (a), mostra-se o comprimento de onda em função de $\log(\text{erro})$ para as cinco densidades que foram obtidas nas Figuras 6.5 (a) - (e). O gráfico da Figura 6.6 (b) mostra o número de elementos produzidos em

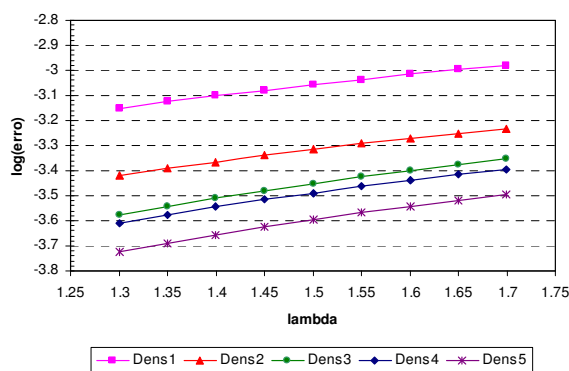
função do $\log(\text{erro})$. Os gráficos da Figura 6.6 (a) e (b) possuem comportamento semelhante ao visto na primeira geometria, de um guia retangular.

lambda	Ge2	dens1	dens2	dens3	dens4	dens5
1,3	1,468504	1,467465	1,467945	1,468114	1,468142	1,468228
1,35	1,466546	1,465448	1,465951	1,466125	1,466156	1,466247
1,4	1,464553	1,463395	1,463921	1,464101	1,464136	1,464231
1,45	1,462526	1,461307	1,461857	1,462042	1,462080	1,462180
1,5	1,460465	1,459185	1,459759	1,459948	1,459991	1,460095
1,55	1,458370	1,457029	1,457627	1,457821	1,457868	1,457977
1,6	1,456242	1,454839	1,455462	1,455660	1,455712	1,455825
1,65	1,454081	1,452617	1,453265	1,453467	1,453523	1,453641
1,7	1,451888	1,450363	1,451035	1,451241	1,451302	1,451425

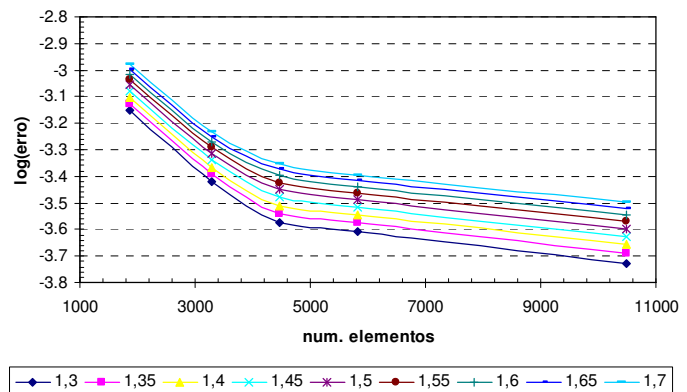
Tab. 6.5: Comprimento de onda e o cálculo do índice modal, do guia tipo costela.

Log(ERRO1)	Log(ERRO2)	Log(ERRO3)	Log(ERRO4)	Log(ERRO5)
-3,150458256	-3,419710896	-3,575616917	-3,607922481	-3,725719342
-3,125632131	-3,391816041	-3,542115486	-3,575614795	-3,69048675
-3,101839027	-3,365204872	-3,510234385	-3,544974282	-3,657211947
-3,079027598	-3,33983759	-3,479927219	-3,515952451	-3,625839098
-3,05715921	-3,315687458	-3,45116094	-3,488516159	-3,596329666
-3,036206496	-3,292739993	-3,423915331	-3,462647253	-3,568662554
-3,016152308	-3,270992628	-3,398183033	-3,438342791	-3,54283489
-2,996989052	-3,250454879	-3,373970162	-3,415615916	-3,518863598
-2,978718427	-3,23114908	-3,351297588	-3,39449751	-3,496787931

Tab. 6.6: Cálculo do $\log(\text{erro})$, do guia tipo costela, em referência ao GiD.



(a)



(b)

Fig. 6.6: (a) Gráfico do $\log(\text{erro})$ em função de λ , (b) Gráfico de queda linear do $\log(\text{erro})$ em função do número de elementos, do guia tipo costela.

6.4 Refinamento de um Guia do Tipo Enterrado

No terceiro teste, é realizada uma análise da constante de propagação em uma geometria de um guia óptico tipo enterrado. Na Figura 6.7 (a) são introduzidas paredes perfeitamente casadas (PML-*Perfectly Matched Layers*) de espessura $2\mu m \times 2\mu m$. A estrutura interna em estudo é composta de um quadrado no centro, de dimensão $2\mu m \times 2\mu m$; dois retângulos, de dimensão $4\mu m \times 1\mu m$, cada um posicionado na região superior e inferior; e dois retângulos $1\mu m \times 2\mu m$, cada um localizado em um dos lados, esquerdo e direito do quadrado central.

A Figura 6.7 (b) mostra as coordenadas da geometria, definidas num plano, vistos em (6.2). O guia interno é composto pelas seguintes geometrias: Quadrado central, $Re t_{sup}$, $Re t_{inf}$, $Re t_{esq}$, $Re t_{dir}$.

$$\text{Quadrado central: } \begin{cases} x = [-1, 1], e y = \pm 1 \\ y = [-1, 1], e x = \pm 1 \end{cases} \quad (6.2)$$

$$Ret_{sup}: \begin{cases} x = [-2, 2], e y = 1 \text{ ou } y = 2 \\ y = [1, 2], e x = \pm 2 \end{cases}, Ret_{inf}: \begin{cases} x = [-2, 2], e y = 1 - \text{ ou } y = -2 \\ y = [-1, -2], e x = \pm 2 \end{cases}$$

$$Ret_{esq}: \begin{cases} x = [-2, -1], e y = \pm 1 \\ y = [-1, 1], e x = -1 \text{ ou } y = -2 \end{cases}, Ret_{dir}: \begin{cases} x = [1, 2], e y = \pm 1 \\ y = [-1, 1], e x = 1 \text{ ou } y = 2 \end{cases}$$

O início da margem superior esquerda da PML tem coordenadas (-6,6) e o início da margem inferior direita da PML tem coordenadas (6,-6).

Na Figura 6.7 (c), quatro tipos de materiais foram utilizados: material 1, material 2, material 3 e material 4. A estrutura interna é composta por três materiais com $n_1=1,5$, $n_2=1,3$ e $n_3=1,45$. O índice de refração externa $n_4=1,00$.

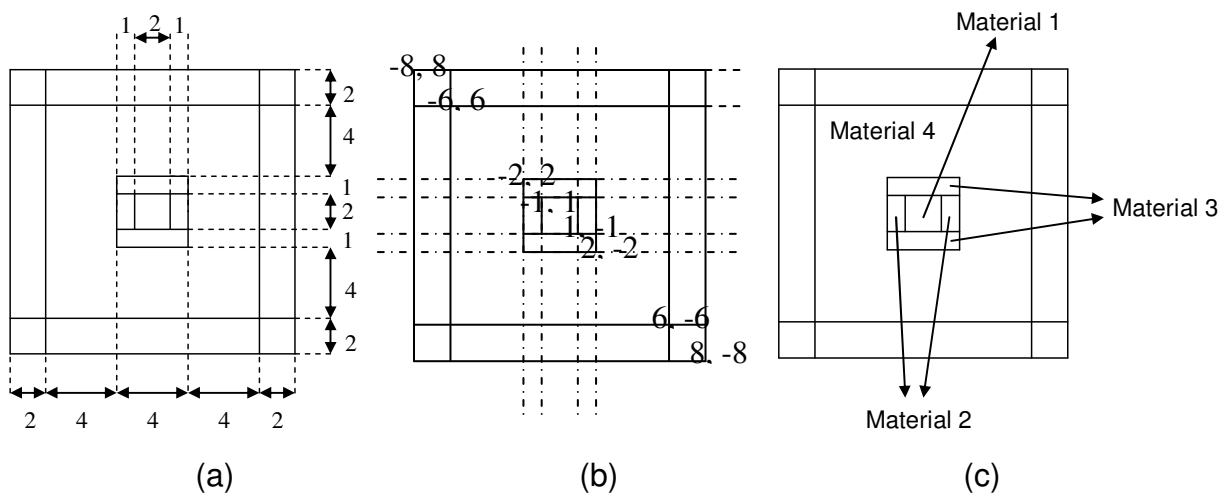


Fig. 6.7: Definindo a (a) proporção do guia tipo enterrado, as (b) coordenadas desejadas, e os (c) tipos de materiais.

Segue-se, na Tabela 6.7, as informações dos cinco testes realizados. E as Figuras 6.8 (a) – (e), mostram a geração de malhas de acordo com os dados dessa Tabela 6.7.

características saída	Num. elem. estimados	Num. elementos	Num. nós	Parâmetros da função gaussiana		
				A	$(x_0; y_0)$	$(d_1; d_2)$
geom3_dens1	1700	1794	931	1,0	(0,0; 0,0)	(2,0; 2,0)
geom3_dens2	3100	3130	1599	1,0	(0,0; 0,0)	(2,0; 2,0)
geom3_dens3	3900	3964	2016	1,0	(0,0; 0,0)	(1,5; 1,5)
geom3_dens4	5500	5524	2796	1,0	(0,0; 0,0)	(2,0; 2,0)
geom3_dens5	10 000	10 048	5058	1,0	(0,0; 0,0)	(2,0; 2,0)

Tab. 6.7: Características de refinamento de um guia tipo enterrado.

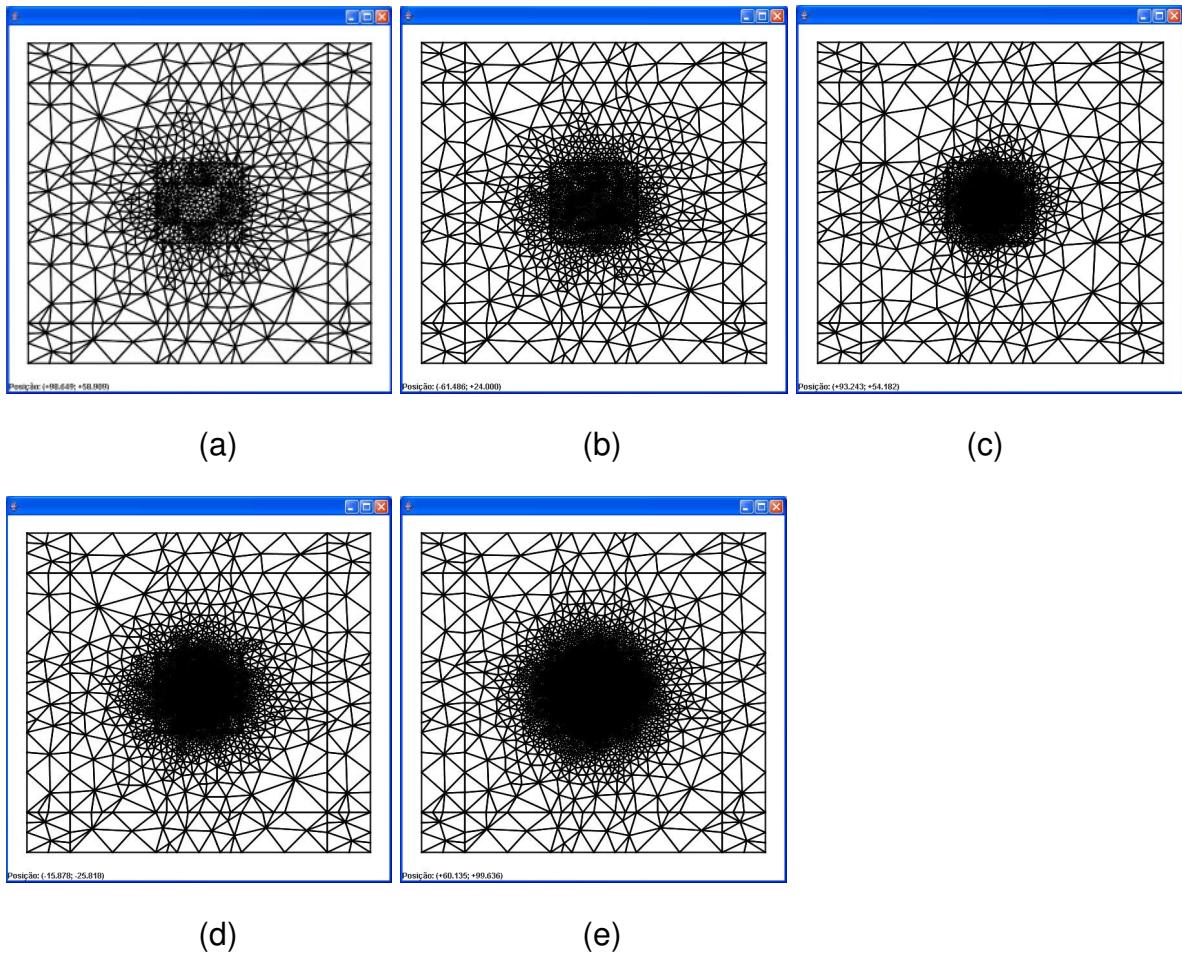


Fig. 6.8: Geração de malhas de um guia tipo costela e as suas seguintes identificações de saída: (a) geom3_dens1, (b) geom3_dens2, (c) geom3_dens3, (d) geom3_dens4, e (e) geom3_dens5.

A Figura 6.9 mostra um mapeamento dos três tipos de materiais da região enterrada da Figura 6.8 (a), diferenciados pelos valores 0, 1 e 2, que são respectivamente, material 1, material 2, e material 3. O quadrado central corresponde ao material 0, os retângulos à esquerda e à direita ao material 1, e os retângulos superior e inferior ao material 2. A delimitação dessas regiões está representada pela linha vermelha.

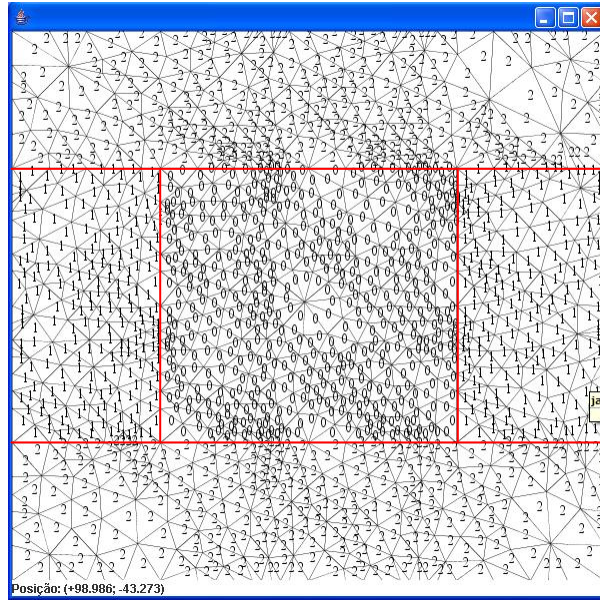


Fig. 6.9: Identificação dos materiais, da malha da Figura 6.8 (a), com zoom em 12x da estrutura interna.

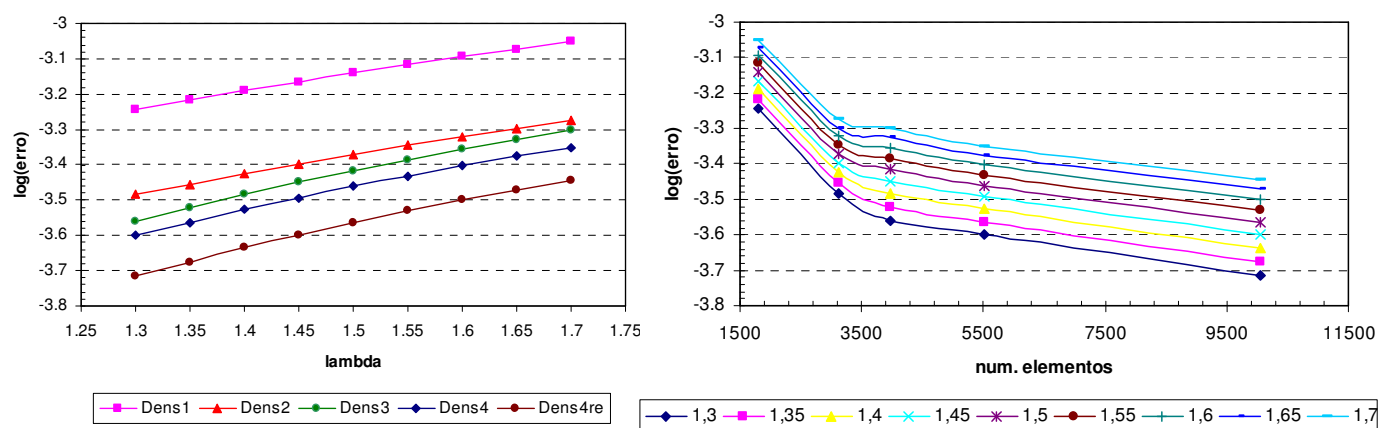
Na Tabela 6.8, em Ge1 produziu-se 17894 elementos e 36145 nós, com simulação de elementos de ordem quadrática, enquanto os demais resultados utilizaram-se ordem linear. De acordo com as Tabelas 6.8 e 6.9 abaixo, obtém-se os gráficos vistos na Figura 6.10 (a) e (b). Em 6.10 (a) mostra-se o comprimento de onda em função de $\log(\text{erro})$ para as cinco densidades obtidas pelas Figuras 6.8 (a) - (e). E o gráfico da Figura 6.10 (b) mostra o número de elementos produzidos em função do $\log(\text{erro})$. Os gráficos 6.10 (a) e (b) gráficos possuem comportamentos semelhantes aos observados na primeira e na segunda geometria.

lambda	Ge3	dens1	dens2	dens3	dens4	dens4re
1,3	1,463487	1,462651	1,463007	1,463083	1,463119	1,463205
1,35	1,461434	1,460546	1,460920	1,460994	1,461034	1,461125
1,4	1,459360	1,458418	1,458811	1,458882	1,458926	1,459022
1,45	1,457265	1,456268	1,456681	1,456748	1,456797	1,456898
1,5	1,455151	1,454097	1,454531	1,454593	1,454648	1,454753
1,55	1,453016	1,451905	1,452360	1,452418	1,452478	1,452588
1,6	1,450863	1,449694	1,450169	1,450222	1,450288	1,450403
1,65	1,448689	1,447462	1,447959	1,448007	1,448079	1,448199
1,7	1,446497	1,445211	1,445730	1,445772	1,445851	1,445976

Tab. 6.8: Comprimento de onda e o cálculo do índice modal, do guia enterrado.

Log(ERRO1)	Log(ERRO2)	Log(ERRO3)	Log(ERRO4)	Log(ERRO4re)
-3,243443836	-3,484372697	-3,559853469	-3,599759748	-3,715935973
-3,216361902	-3,454249867	-3,521517666	-3,562572681	-3,674649139
-3,190127898	-3,425236542	-3,484891189	-3,527111656	-3,635494057
-3,164747155	-3,397345575	-3,449949023	-3,493360994	-3,598425194
-3,140229132	-3,370594907	-3,416669525	-3,461308341	-3,563404402
-3,116586704	-3,345007299	-3,38503525	-3,430945777	-3,530402113
-3,093835987	-3,320610712	-3,355034144	-3,402271332	-3,499398964
-3,071996659	-3,297439346	-3,326661122	-3,375290987	-3,470387997
-3,051092788	-3,275535425	-3,299920178	-3,350021301	-3,44337762

Tab. 6.9: Cálculo de $\log(\text{erro})$, do guia tipo enterrado, em referência ao GiD.



(a)

(b)

Fig. 6.10: (a) Gráfico de $\log(\text{erro})$ em função de λ , (b) Gráfico de queda linear de $\log(\text{erro})$ em função do número de elementos, do guia tipo enterrado.

Capítulo 7

Conclusões e Trabalhos Futuros

7.1 Conclusões

Nesta tese, descreveu-se o processo de geração de malhas através das técnicas baricêntrica, triangulação de Delaunay, e a suavização laplaciana. Estudou-se, também, a geração de malhas com o intuito de manter a preservação dos contornos para as interfaces retangulares, circulares e curvas arbitrárias. Tal integridade é respeitada através do cálculo do *stitch point*, ou seguindo a denominação utilizada neste trabalho, de ponto de recuperação. De acordo com essa base desenvolvida, a integridade de bordas poderá ser aperfeiçoada nos trabalhos futuros, como, por exemplo, a automatização para um conjunto de interfaces bidimensionais, irregulares e distintas.

Outro resultado alcançado, através da implementação das técnicas apresentadas, é o re-malhamento adaptativo, onde as vantagens são evidentes quando tratamos de uma seqüência de cálculos, ou seja, quando se busca continuamente uma solução, cujo parâmetro esteja sempre variando ou o foco de interesse se move continuamente no domínio.

De forma geral, a geração de malha apresentada, neste trabalho, mostra a coerência da qualidade dos elementos obtidos, em relação aos trabalhos desta mesma área, tais como [2,5].

Por fim, devido à característica da programação orientada a objeto e da documentação dos artefatos gerados, reserva-se outras características, frutos da implementação: a funcionalidade, a facilidade na manutenção, e a expansão do projeto.

7.2 Trabalhos Futuros

Existem diversos assuntos que podem ser acrescentados à biblioteca, no sentido de expandir as funcionalidades do programa. Primeiramente, existe a possibilidade de inserção dos algoritmos dentro de um ambiente visual robusto e integrado ao OpenGL, estudo realizado por Malavazi [4], que são recursos utilizados em ambientes amigáveis, como o GiD [35] e o Gmesh [44].

Existe uma grande atenção voltada à integração de técnicas de processamento de imagens digitais bidimensionais, como a biblioteca JAI [45] para Java, com o objetivo de segmentar os buracos de fibras fotônicas. Essa discretização permitirá efetuar um pré-processamento sobre as interfaces, de modo a produzir um refinamento adequado.

Também é interessante partir para o conhecimento de outras técnicas de refinamento, como a aplicação do avanço de fronteira [26], que também pode ser agregada ao Delaunay, abordagem bastante complexa, entretanto bastante interessante para o aprendizado, e que se tem destacado em programas como o *Triangle* [46].

Outro assunto é realizar testes de desempenho de geração de malhas considerando o processamento paralelo.

Tem-se, também, a possibilidade de estudar erros de discretização provocados por arredondamentos de operações matemáticas processadas no computador que pode determinar a falta de precisão nos retornos das rotinas de localização nodal, como, por exemplo, se um nó está dentro ou fora do circuncírculo de uma possível triangulação de Delaunay.

Referências Bibliográficas

- [1] Gerald Farin, *Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide*, 5ª Edição, Editora Morgan-Kaufman, 2001.
- [2] Emir Baude, “Refinamento Adaptativo e Malhas 2D e 3D para a Solução Numérica de Problemas de Propagação Eletromagnética,” tese de doutorado, Departamento de Microonda e Óptica (DMO), Faculdade de Engenharia Elétrica e de Computação (FEEC), Universidade Estadual de Campinas (UNICAMP), maio 2000.
- [3] Adriano C. Lisboa, “Geração de Malhas para Geometrias Curvas,” tese de doutorado, Centro de Pesquisa e Desenvolvimento em Engenharia Elétrica (CPDEE), Departamento de Engenharia Elétrica, Universidade Federal de Minas Gerais (UFMG), junho 2005.
- [4] Karina Malavazi, “Aspectos de Usabilidade Aplicados a Sistemas Geradores de Malhas,” Notas de leitura da tese de mestrado, Departamento de Microonda e Óptica (DMO), Faculdade de Engenharia Elétrica e de Computação (FEEC), Universidade Estadual de Campinas (UNICAMP), março 2007.
- [5] Manuel Ribeiro Filho, “Geração Automática e Adaptativa de Malhas para Elementos Finitos 2D com Aplicações em Eletromagnetismo,” tese de doutorado, Centro Tecnológico (CT), Programa de Pós-Graduação em Engenharia Elétrica (PPGEE), Universidade Federal do Pará (UFPA), junho 2002.

- [6] Vicente H. F. Batista, "Geração de Malhas Não-Estruturadas Tetraédricas Utilizando um Método de Avanço de Fronteira," dissertação de mestrado, Centro de Tecnologia, Universidade Federal do Rio de Janeiro (UFRJ), junho 2005.
- [7] Mark T. Jones; Paul E. Plassmann, "Adaptive refinement of unstructured finite-element meshes," *Finite Elements in Analysis and Design*, 25, pp. 41–60, 1997.
- [8] C. Bryant, "Two dimensional automatic triangular mesh generation," *IEEE Transactions on Magnetics*, vol. 21, n° 6, pp. 2547- 2550, 1985.
- [9] Mark Shephard, "Automatic And Adaptive Mesh Generation," *IEEE Transactions on Magnetics*, vol. 21, n° 6, pp. 2484- 2489, 1985.
- [10] B. K. Karamete, T. Tokdemir, e M. Ger, "Unstructured Grid Generation and Simple Triangulation Algorithm For arbitray 2-D Geometry Using Object Oriented Programing," *International Journal for Numerical Methods in Engineering*, vol. 40, n° 2, pp. 251-268, 1997.
- [11] R. V. Nambiar, R. S. Valera, e K. L. Lawrence, "An Algorithm for Adaptive Refinement of Triangular Element Meshes," *International journal for numerical methods in engineering*, vol. 36, n° 3, pp. 499-509, 1993.
- [12] J. R. Shewchuk, "*Delaunay Refinement Mesh Generation*," phd thesis, School of Commputer Science, Computer Science Depatment, Carnegie Mellon University, Pittsburgh, maio 1997.
- [13] F. A. Fernandez, Y. C. Yong, e R. D. Ettinger, "A simple Adaptive Mesh Generator for 2-D Finite Element Calculations," *IEEE Transactions On Magnetics*, vol. 29, n° 5, pp.0018-9464, 1993.
- [14] P. Fernandes, P. Girdinio, M. Repetto, e G. Secondo, "Refinement strategies in adaptive meshing," *IEEE Trans. on Magnetics*, vol. 28, n° 2, pp. 1739-1742, 1992.

- [15] Guiseppe Pelosi, Roberto Coccioli, e Stefano Selleri, “*Quick Finite Elements for Electromagnetic Waves*,” Editora Artech House Publishers, 1^a Edição, 256 páginas, Londres, junho 1998.
- [16] M. C. Rivara, “Selective Refinement/Derefinement Algorithms For Sequences of Nested Triangulations,” *International Journal for Numerical Methods in Engineering*, vol. 28, n^o 12, pp. 2889-2906, 1989.
- [17] M. C. Rivara, “New Longest-edge algorithms for the refinement and/or improvement of unstructured Triangulations,” *International Journal for Numerical Methods in Engineering*, vol. 40, n^o 18, pp. 3313-3324, 1997.
- [18] M. C. Rivara e P. INOSTROZA, “Using Longest-side bisection techniques for the automatic refinement of Delaunay Triangulations,” *International Journal for Numerical Methods in Engineering*, vol. 40, n^o 4, pp. 581-597, 1997.
- [19] R. J. Shewchuk, “Lecture Notes on Delaunay Mesh Generation,” tese de doutorado, Department of Electrical Engineering and Computer Science, University of California at Berkeley, 1999.
- [20] Shang-Hua Teng, “Computational Geometry Lecture Three,” tese de doutorado, Department of Computer Science, Boston University, outubro 2004.
- [21] J. F. Lee e R. Dyczij-Edlinger, “Automatic Mesh Generator Using a Modified Delaunay Tessellation,” *IEEE Antennas and Propagation Magazine*, vol. 39, n^o 1, pp.1045-9243, 1997.
- [22] H. Borouchaki, F. Hecht, E. Saltel, e P. L. Gerge, “Reasonably efficient Delaunay Based mesh generator in 3 dimensions,” *Proceedings 4th International Meshing Roundtable*, Sandia National Laboratories, pp.3-14, outubro 1995.
- [23] Marina Gavrilova. “Proximity and Applications in General Metrics,” tese de doutorado, Dept. of Computer Science and Dept. of Mechanical Engineering, University of Calgary, Alberta, 1998.

- [24] J. R. Shewchuk, "What is a good linear finite element? Interpolation, Conditioning, anisotropy, and Quality Measures," Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, dezembro 2002.
- [25] J. O'Dwyer e P. Evans, "Triangular Element Refinement in Automatic Adaptive Mesh Generation," *IEEE Transactions on Magnetics*, vol. 33, nº 2, pp. 1740-1743, 1997.
- [26] Philippe P. Pebay e Timothy J. Baker, "Analysis of triangle quality measures," *Mathematics of Computation*, vol. 72, nº 244, pp. 1817–1839, 2003.
- [27] Marcos A. R. Franco, Nancy Mieko Abe; Angelo Pássaro, Francisco Sircilli, Valdir Augusto Serrão, Diogo Hideo Odan; Guilherme Moreira Magalhães, e F. J. R. SANTOS. Análise Computacional de Dispositivos e Componentes de Óptica Integrada, Fibras Ópticas e Guias de Microondas. In: III Simpósio de Engenharia Inercial - III SBEIN, 2001, Rio de Janeiro. Anais do III Simpósio de Engenharia Inercial - III SBEIN, vol. 1, 2001.
- [28] John Liu e Murray R. Spiegel, *Manual de Fórmulas e Tabelas Matemáticas*. Editora McGraw-Hill, 1ª Edição, 270 páginas, 1992.
- [29] C. K. Lee e R. E. Hobbs, "Automatic Adaptive Finite Element Mesh Generation over Arbitrary Two-Dimensional Domain Using Advancing Front Technique," *Computers & Structures*, vol. 71, pp. 9-34, 1998.
- [30] T. Zhou e K. Shimada. "An Angle-Based Approach to Two-Dimensional Mesh Smoothing," *Proceedings 9th International Meshing Roundtable*, Sandia National Laboratories, pp.373-384, outubro 2000.
- [31] L. A. Freitag, "On Combining Laplacian And Optimization-Based Mesh Smoothing Techniques," *AMD Trends in Unstructured Mesh Generation*, ASME, vol. 220, pp.37-43, julho 1997.

- [32] H. Borouchaki e P. L. George, "Aspects of 2-D Delaunay Mesh Generation," *International Journal for Numerical Methods In Engineering*, Wiley, vol. 40, n° 11, pp.1957-1975, 1997.
- [33] W. H. Frey, "Selective Refinement: A New Strategy For Automatic Node Placement in Graded Triangular Meshes," *International Journal for Numerical Methods in Engineering*, vol. 24, n° 11, pp. 2183-2200, 1987.
- [34] W. H. Frey, "Mesh Relaxation: A New Technique For Improving Triangulation," *International Journal for Numerical Methods in Engineering*, vol. 31, n° 31, pp. 1121-1133, 1991.
- [35] GiD, Página da internet, fevereiro 2007. <http://gid.cimne.upc.es/>.
- [36] Mariane F. Galhardo e Luciana M. Zaina, "Simulação para Ensino de Conceitos da Orientação a Objetos," *XIII SEMINCO (Seminário de Computação)*, Universidade Regional de Blumenau, 2004.
- [37] Rafael Santiago e Rudimar L. S. Dazzi, "Ferramenta de apoio ao ensino de algoritmos," *XIII SEMINCO (Seminário de Computação)*, Universidade Regional de Blumenau, 2004.
- [38] Y. Xiang, Y. Ping, e W. Zhou, "Object-Oriented Finite Element Analysis: A Distributed Approach to Mesh Generation," *The 5th International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP 2002)*, IEEE Press, pp. 92-99, Beijing, China, outubro 2002.
- [39] Maria C. Bastarrica e N. Hitschfeld-Kahler, "An Evolvable Meshing Tool Through a Flexible Object-Oriented Design," *Proceedings, 13th International Meshing Roundtable*, pp.203-212, 2004.
- [40] H. M. Deitel e P. J. Deitel, *Java: Como Programar*, Editora Prentice-Hall, 5ª Edição, 1152 páginas, 2005.

- [41] José Pissolato Filho, Notas de leitura de EE754 em Ondas guiadas, Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas.
- [42] Eclipse, Página da internet, fevereiro 2007. <http://www.eclipse.org/downloads/>.
- [43] Netbeans, Página da internet, fevereiro 2007. <http://www.netbeans.org/>.
- [44] C. Geuzaine e J. Remacle, "Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities", Página da internet, fevereiro 2007, <http://geuz.org/gmsh/>.
- [45] "JAI: Java Advanced Imaging API", Página da internet, fevereiro 2007. <http://java.sun.com/javase/technologies/desktop/media/>.
- [46] J. R. Shewchuk, "Triangle: A two-Dimensional Quality Mesh Generator and Delaunay Trinagulator", Página da internet, fevereiro 2007, <http://www.cs.cmu.edu/~quake/triangle.html>.
- [47] G. R. Liu e S. S. Quek, *"The Finite Element Method - A practical course,"* Editora Butterworth-Heinemann, 384 páginas, 2003.
- [48] Vitaly Félix Rodríguez-Esquerre, "Modelagem de estruturas fotônicas por elementos finitos 2D no domínio do tempo," tese de doutorado, Departamento de Microonda e Óptica (DMO), Faculdade de Engenharia Elétrica e de Computação (FEEC), Universidade Estadual de Campinas (UNICAMP), 2003.
- [49] Atílio C. Hsu, Carlos Henrique S. Santos, e Vinicius J. Latorre, "Desenvolvimento de um Simulador Cruise Control Utilizando a Metodologia Híbrida XP@SCRUM e a Semiótica Organizacional", Relatório Final de Atividades, Instituto de Computação (IC), Universidade Estadual de Campinas, 2005.

Apêndice A

Método dos Elementos Finitos

A.1 Método dos Elementos Finitos Aplicado à Resolução da Equação de Poisson

A equação de Poisson, obtido em [47], dada por (A.1),

$$\nabla^2 \phi = \frac{\rho}{\varepsilon} \quad (\text{A.1})$$

estabelece uma equação diferencial para o potencial elétrico ϕ , numa região de um espaço de volume V , delimitada por uma superfície S .

Se $\phi_1, \phi_2, \dots, \phi_n$ forem soluções linearmente independentes, então a solução geral é a soma de todas as soluções, com coeficientes apropriados, ou seja, $\phi = \alpha_1 \phi_1 + \alpha_2 \phi_2 + \dots + \alpha_n \phi_n$.

Portanto, através do princípio de superposição tem-se que:

$$\phi = \sum_{j=1}^N \alpha_j \phi_j \quad (\text{A.2})$$

Supõe-se que ϕ , definido em (A.2), seja a solução aproximada. Em seguida, substituindo ϕ em (A.1), tem-se um resíduo R diferente de zero, mostrado em (A.3).

$$R(\phi) = \left(\sum_{j=1}^N \alpha_j \phi_j \right) - \frac{\rho}{\varepsilon} \neq 0 \quad (\text{A.3})$$

A melhor aproximação para ϕ é a que reduz o resíduo R para valores mínimos em todos os pontos do domínio r . Dessa forma, o método dos resíduos ponderados impõe a seguinte condição,

$$\int_r R(\phi) w dr = 0 \quad (\text{A.4})$$

onde w é a função de ponderação.

Substituindo-se o valor de $R(\phi)$ em (A.4) obtém-se (A.5):

$$\int_r \left(\nabla^2 \phi - \frac{\rho}{\varepsilon} \right) \phi_i dr = 0 \quad (\text{A.5})$$

Resolvendo a expressão (A.5), obtém-se (A.6):

$$\sum \alpha_j \left(\int_r \nabla^2 \phi_j \phi_i dr \right) - \int_r \frac{\rho}{\varepsilon} \phi_i dr = 0 \quad (\text{A.6})$$

A primeira indentidade de *Green* estabelece a seguinte equação:

$$\int_v \psi \nabla^2 \phi + \nabla \phi \cdot \nabla \psi dv = \oint_s \phi \nabla \psi \cdot \hat{n} dA \quad (\text{A.7})$$

Aplicando ela para as seguintes funções $\phi = \psi$ de (A.7), obtém-se (A.8):

$$\int_v (\phi \nabla^2 \phi + \nabla \phi \cdot \nabla \phi) d\phi = \oint_s \phi \nabla \phi \cdot \hat{n} dA \quad (\text{A.8})$$

Como ϕ satisfaz a equação de *Laplace*, tem-se (A.9):

$$\int_v |\nabla \phi|^2 dV = \oint_s \phi \frac{\partial \phi}{\partial n} dA \quad (\text{A.9})$$

A partir de (A.6), e usando a primeira identidade de *Green*, obtém-se (A.10):

$$\sum \alpha_j \left[- \int_r \nabla \phi_j \cdot \nabla \phi_i dr + \oint_{\Gamma} \frac{\partial \phi_j}{\partial n} \cdot \phi_i d\Gamma \right] = \int_r \frac{\rho}{\varepsilon} \phi_i dr \quad (\text{A.10})$$

O conjunto de equações pode ser reescrito utilizando uma notação matricial da forma $A \cdot \vec{\alpha} = \vec{b}$:

$$A_{ij} = - \int_r \nabla \phi_j \cdot \nabla \phi_i dr + \oint_{\Gamma} \frac{\partial \phi_j}{\partial n} \cdot \phi_i d\Gamma \quad (\text{A.11})$$

$$b_i = \int_r \frac{\rho}{\varepsilon} \phi_i dr \quad (\text{A.12})$$

$$\vec{\alpha} = \begin{pmatrix} a_1 \\ \vdots \\ a_N \end{pmatrix} \quad (\text{A.13})$$

onde o sistema matricial apresentado corresponde a um determinado elemento e a matriz A é simétrica.

Portanto, o método dos resíduos ponderados substitui a solução aproximada na equação a ser resolvida, obtendo-se um resíduo que, multiplicado por uma função de ponderação, é minimizado resultando na matriz elementar.

A.2 Caso Bidimensional

De acordo com Rodríguez-Esquerre [48], a função desconhecida, ϕ , dentro de cada elemento triangular, pode ser aproximada como:

$$\phi^e(x, y) = a^e + b^e x + c^e y \quad (\text{A.14})$$

onde a^e , b^e e c^e são coeficientes a serem determinados e e é o número do elemento.

Nos elementos lineares tem-se três nós localizados nos vértices dos triângulos, como visto na Figura A.1. Os nós são numerados em sentido anti-horário com os números 1, 2 e 3, com os correspondentes valores de ϕ denotados por ϕ_1^e , ϕ_2^e e ϕ_3^e . Aplicado em (A.14), obtém-se:

$$\begin{aligned} \phi_1^e(x, y) &= a_1^e + b_1^e x_1^e + c_1^e y_1^e \\ \phi_2^e(x, y) &= a_2^e + b_2^e x_2^e + c_2^e y_2^e \\ \phi_3^e(x, y) &= a_3^e + b_3^e x_3^e + c_3^e y_3^e \end{aligned} \quad (\text{A.15})$$

Resolvendo-se este sistema para os coeficientes constantes a^e , b^e e c^e em termos de ϕ_j^e , e substituindo em (A.15), tem-se (A.16):

$$\phi^e(x, y) = \sum_{j=1}^3 N_j^e(x, y) \phi_j^e \quad (\text{A.16})$$

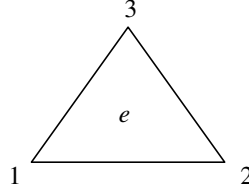


Fig. A.1: Elemento triangular linear.

onde $N_j^e(x, y)$ representa as funções de base ou de interpolação e são dadas por:

$$N_j^e(x, y) = \frac{1}{2A^e} (a_j^e + b_j^e x + c_j^e y) \quad (\text{A.17})$$

onde $j=1, 2, 3$ e A^e é a área do elemento e . Os demais parâmetros são dados por interpolação e são dados por:

$$\begin{aligned} a_1^e &= x_2^e y_3^e - y_2^e x_3^e & b_1^e &= y_2^e - y_3^e & c_1^e &= x_3^e - x_2^e \\ a_2^e &= x_3^e y_1^e - y_3^e x_1^e & b_2^e &= y_3^e - y_1^e & c_2^e &= x_1^e - x_3^e \\ a_3^e &= x_1^e y_2^e - y_1^e x_2^e & b_3^e &= y_1^e - y_2^e & c_3^e &= x_2^e - x_1^e \end{aligned} \quad (\text{A.18})$$

e A^e é calculado através de (A.19):

$$A^e = \frac{1}{2} \begin{vmatrix} 1 & x_1^e & y_1^e \\ 1 & x_2^e & y_2^e \\ 1 & x_3^e & y_3^e \end{vmatrix} = \frac{1}{2} (b_1^e c_2^e - b_2^e c_1^e) \quad (\text{A.19})$$

Nas expressões anteriores, x_j^e e y_j^e , para $j=1,2,3$, denotam os valores das coordenadas do j -ésimo nó do elemento e .

É facilmente demonstrado que as funções de base têm a seguinte propriedade, vista em (A.20),

$$N_j^e(x_j^e, y_j^e) = \delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \quad i, j = 1, 2, 3 \quad (\text{A.20})$$

onde δ_{ij} é o *Delta de Kronecker*, ou seja, as funções de base assumem o valor 1 quando os índices $i = j$ coincidem, e 0 para outro caso qualquer.

A.3 Matriz Global

Nesta seção é construído o sistema matricial envolvendo todos os pontos nodais, que também é denominado de sistema global. A dimensão das matrizes globais é determinada pelo número de nós da malha. Com o objetivo de descrever esse método, apresenta-se uma malha simples, mostrada na Figura A.2 (a), que é composta de seis nós. Esses nós definem a numeração global da malha, enquanto que na Figura A.2 (b), ilustra-se a padronização da numeração local.

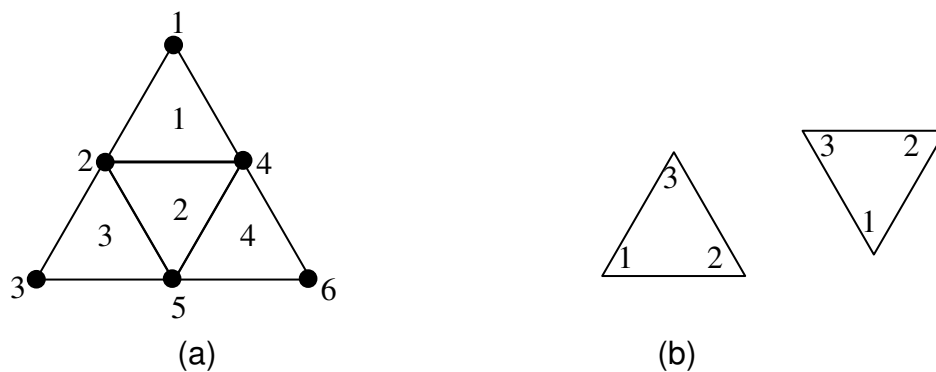


Fig. A.2: Elemento triangular linear. (a) Exemplo arbitrário de malha 2-D, (b) Padrão de numeração local dos nós dos elementos.

A Tabela A.1 ilustra a associação entre a identificação do elemento e as suas numerações locais. Por exemplo, de acordo com a Figura A.2 (a), o elemento 1 cuja numeração local do vértice é 1, corresponde à numeração global 2.

Num. Local	Num. Elemento			
	1	2	3	4
1	2	5	3	5
2	4	4	5	6
3	1	2	2	4

Tab. A.1: Matriz que associa o elemento com sua numeração local e global.

A Tabela A.2 mostra a tabela do elemento associado ao parâmetro a_{ij}^e .

	Num. Elemento											
	1			2			3			4		
Num. Local	1	2	3	1	2	3	1	2	3	1	2	3
1	a_{11}^1	a_{12}^1	a_{13}^1	a_{11}^2	a_{12}^2	a_{13}^2	a_{11}^3	a_{12}^3	a_{13}^3	a_{11}^4	a_{12}^4	a_{13}^4
2	a_{21}^1	a_{22}^1	a_{23}^1	a_{21}^2	a_{22}^2	a_{23}^2	a_{21}^3	a_{22}^3	a_{23}^3	a_{21}^4	a_{22}^4	a_{23}^4
3	a_{31}^1	a_{32}^1	a_{33}^1	a_{31}^2	a_{32}^2	a_{33}^2	a_{31}^3	a_{32}^3	a_{33}^3	a_{31}^4	a_{32}^4	a_{33}^4

Tab. A.2: Matriz global, com a inclusão da primeira matriz elementar.

Finalmente, a Tabela A.3 mostra o preenchimento da matriz global a partir das matrizes de cada elemento. A numeração local de cada elemento e a numeração global é fundamental para a montagem correta do sistema global.

	1	2	3	4	5	6
1	a_{33}^1	a_{31}^1	0	a_{32}^1	0	0
2	a_{13}^1	$a_{11}^1 + a_{33}^2$	a_{31}^3	$a_{12}^1 + a_{32}^2$	$a_{31}^2 + a_{32}^3$	0
3	0	a_{13}^3	a_{11}^3	0	a_{12}^3	0
4	a_{23}^1	$a_{21}^1 + a_{23}^2$	0	$a_{22}^1 + a_{22}^2 + a_{33}^4$	$a_{21}^2 + a_{31}^4$	a_{32}^4
5	0	$a_{13}^2 + a_{23}^3$	a_{21}^3	$a_{12}^2 + a_{13}^4$	$a_{11}^2 + a_{22}^3 + a_{11}^4$	a_{12}^4
6	0	0	a_{23}^4	a_{23}^4	a_{21}^4	a_{22}^4

Tab. A.3: Matriz global resultante.

Apêndice B

Simplexos e Padrões de Numeração

Neste apêndice, realiza-se uma descrição do conceito de simplexos, e os padrões de numeração local e global dos elementos utilizados na geração de malhas.

B.1 Simplexos

De acordo com Lisboa [3], dado um espaço n -dimensional, um simplexo é formado por $n+1$ vértices $v_i \in R^n, i = 1, \dots, n+1$, onde todos eles estão conectados entre si. O número N de arestas é dado pela combinação de $n+1$ vértices, tomados dois a dois, visto da seguinte forma:

$$N_A = \binom{n+1}{2} = \frac{n(n+1)}{2} \quad (\text{B.1})$$

Dessa forma, o simplexo na primeira dimensão é um segmento, na segunda um triângulo, e na terceira um tetraedro.

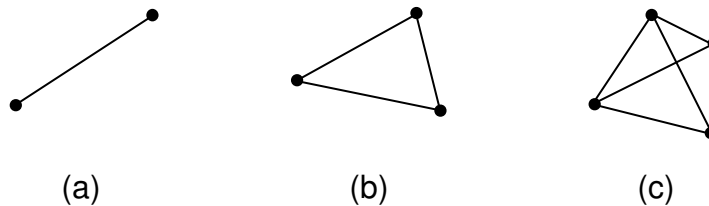


Fig. B.1: Simplexos para (a) $n = 1$, (b) $n = 2$ e (c) $n = 3$.

As facetas dos simplexos possuem n vértices. O número de facetas de um simplexo é dado pela combinação de $n + 1$ elementos tomados n a n ,

$$N_v = \binom{n+1}{n} = n+1 \quad (\text{B.2})$$

Elementos com arestas de comprimentos iguais são chamados de simplexos regulares.

$$L = \|v_i - v_j\| = cte, i \neq j, \text{ para } i, j = 0 \dots, n \quad (\text{B.3})$$

B.2 Padrões de Numeração

A numeração dos nós deve possuir uma identificação única e sem repetição. A Figura B.2 (a) ilustra um exemplo dessa numeração, que está representada pelos nós 1, 2, 3 e 4. Convém denominarmos eles de numeração global dos nós.

Cada nó gerado deve possuir uma identificação. Dessa forma, como pode ser visto na Figura B.2 (b), após a aplicação da técnica baricêntrica, deve-se gerar dois novos nós, que são: 5 e 6.

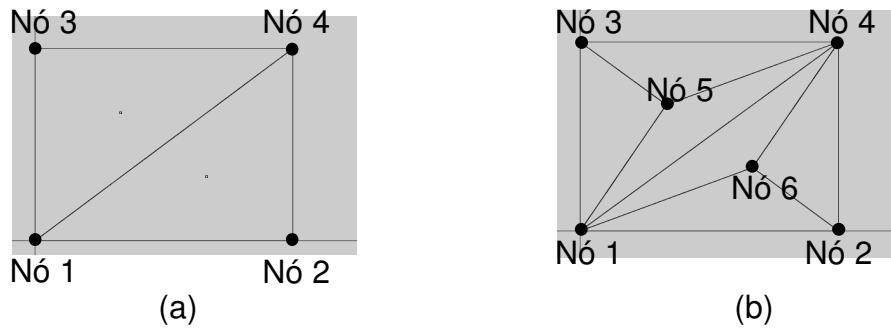


Fig. B.2: Padrão de numeração dos nós globais, (a) antes, com quatro nós de identificação, e (b) após a técnica baricêntrica, produzindo-se seis nós.

Na Figura B.3, observa-se a numeração local dos nós. A numeração local deve ser realizada para cada elemento. Portanto, na configuração inicial, visto na Figura B.3 (a), tem-se apenas dois elementos. Com a aplicação da técnica

baricêntrica, observada na Figura B.3 (b), tem um total de seis elementos. Tanto na Figura B.3 (a), quanto na Figura B.3 (b), todos os elementos devem estar numerados de 1 a 3. Essa numeração deve ser realizada, por convenção, no sentido anti-horário.

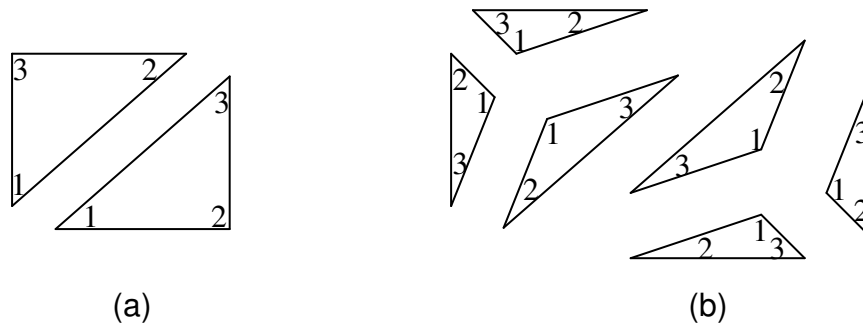


Fig. B.3: Padrão de numeração dos nós locais. (a) antes, com a numeração dos elementos iniciais, e (b) após a técnica baricêntrica.

A numeração dos triângulos deve ser efetuada de forma única, e sem sobreposição. Para cada aplicação da técnica baricêntrica, sobre um elemento, é necessário sempre manter a numeração antiga em um dos novos, e gerar duas novas identificações. Como pode ser visto na Figura B.4 (a), tem-se dois elementos numerados de 1 a 2, e na Figura B.4 (b), após a técnica baricêntrica, resulta seis novos elementos, numerados de 1 a 6. Por exemplo, o elemento 1 da Figura B.4 (a) produz os elementos com numeração 1, 3, 4, mostrado na Figura B.4 (b).

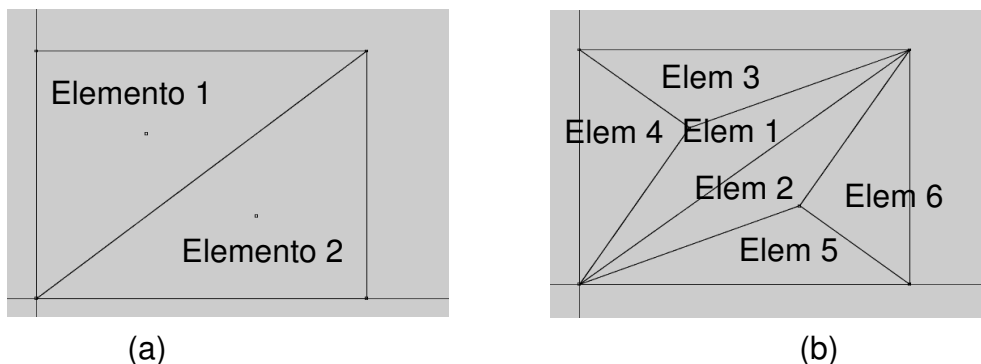


Fig. B.4: Numeração dos elementos, (a) antes, com malha inicial de dois elementos, e (b) após, produzindo-se seis elementos.

Existem dois tipos de numeração para os elementos: global e local. O primeiro é responsável pela identificação geral do elemento na malha. Cada elemento está associado a três numerações locais, assumindo que a disposição delas encontra-se no sentido anti-horário. Dessa forma, pode-se obter a identificação nodal através da numeração global, combinada com qualquer numeração local de 1 a 3.

Apêndice C

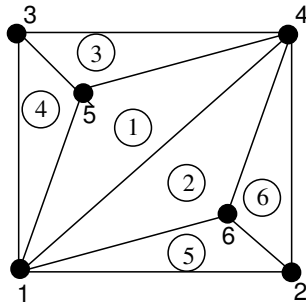
Descrição dos algoritmos

Nesta seção descrevem-se detalhadamente algumas partes do algoritmo desenvolvido. Além disso, procurou-se solucionar problemas através de códigos otimizados de forma a obter um tempo de processamento satisfatório.

C.1 Algoritmo de Busca de Vizinhos Adjacentes

A detecção de vizinhos é um dos principais algoritmos de um programa complexo. Neste apêndice será abordada apenas a questão ilustrativa. O processo de pesquisar e atualizar os atributos de um elemento ocorre em todo o laço de repetição, o que o torna relevante principalmente no desempenho computacional do sistema. O tempo de execução de uma simulação depende bastante da escolha do algoritmo utilizado.

A Fig. C.1 (a) ilustra uma malha de entrada que passará por um processo de refinamento. A numeração global dos nós está localizada próximo aos vértices enquanto a numeração dos elementos está representada por um contorno circular. A Fig. C.1 (b) mostra as informações de mapeamento dos elementos ligados a um nó, ou seja, é efetuada uma associação de cada ponto em comum com os elementos da malha. Nessa listagem, por exemplo, o ponto 1 possui os elementos 1, 2, 4 e 5 ligados a ele.



(a)

Nó	Elementos Ligados					
1	1	2		4	5	
2					5	6
3			3	4		
4	1	2	3			6
5	1		3	4		
6		2			5	6

(b)

Fig. C.1: (a) Malha de entrada. (b) Mapeamento de elementos ligados ao ponto.

É necessário ainda possuir um registro de todos os vértices que compõem cada elemento. A Tabela C.1 mostra um exemplo dos quatro elementos escolhidos da malha da Figura C.1 (a).

Elemento1	Nó 1 (0,0; 0,0)	Nó 4 (40,0; 40,0)	Nó 5 (13,33; 26,66)
Elemento2	Nó 1 (0,0; 0,0)	Nó 6 (26,66; 13,33)	Nó 4 (40,0; 40,0)
Elemento3	Nó 5 (13,33; 26,66)	Nó 4 (40,0; 40,0)	Nó 3 (0,0; 40,0)
Elemento4	Nó 5 (13,33; 26,66)	Nó 3 (0,0; 40,0)	Nó 1 (0,0; 0,0)

Tab. C.1: Identificação dos nós e suas coordenadas, para um dado elemento.

Através das informações da Tabela C.1, toma-se como exemplo o elemento 1 que é composto pelos vértices 5,1, e 4. A Fig. C.2 mostra um exemplo de detecção de vizinho para o lado $\overline{51}$ do elemento 1. Para o lado $\overline{51}$ a intersecção dos elementos, excluindo-se o elemento corrente, é o elemento 4.

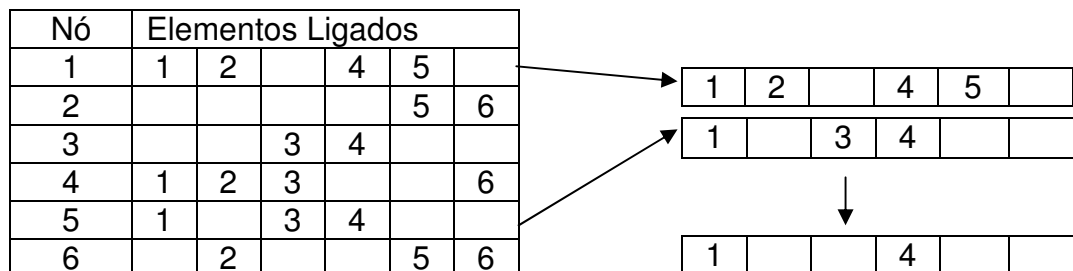


Fig. C.2: Detecção de vizinho através da intersecção dos elementos ligados aos nós 1 e 5.

Para o lado $\overline{14}$, o elemento 2, e o lado $\overline{45}$, o elemento 3. Dessa forma, foi possível buscar os vizinhos do elemento 1, que são 4, 2 e 3.

O algoritmo utilizado é relativamente mais rápido do que percorrer uma lista inteira na pesquisa dos elementos adjacentes. No entanto, existem outras formas otimizadas de armazenar e pesquisar elementos vizinhos de um dado elemento.

C.2 Descrição dos Algoritmos de Integridade

Para quaisquer elementos pertencentes à interface, o elemento deve estar disposto de forma que a numeração local 2 e 3 sejam nós de interface ordenados sempre no sentido anti-horário.

Através da Figura C.3 pode-se buscar facilmente a identificação dos dois nós do contorno. Essa disposição simplifica a implementação, evitando a necessidade de se pesquisar quais vértices dos elementos estão posicionados na interface.

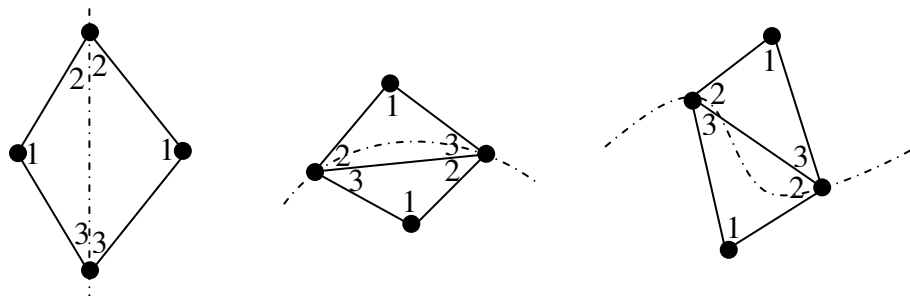


Fig. C.3: Disposição da numeração local em elementos de interfaces distintas.

Os algoritmos de integridade das interfaces estão descritas nas próximas seções.

C.2.1 Algoritmo de Integridade de Interfaces Retangulares

Para interfaces retangulares, o algoritmo pode ser observado na Tabela C.2, onde $Elem_{interface}$ é o vetor que armazena todos os elementos que fazem parte da interface; $N_{elemento}$ é o número total de elementos gerados.

Algoritmo: Integridade em interface retilínea e retangular
 Entrada: $Elem_{interface} = \{h_1, h_2, \dots, h_n\}$

1. Para elemento = 1... $N_{elemento}$
2. Se $elemento \in Elem_{interface}$
3. Se elemento possui par oposto que $\in Elem_{interface}$
4. obter pontos do $elemento = p_1, p_2, p_3$
5. obter pontos do $elemento_{oposto} = po_1, po_2, po_3$
6. calcular $p' = (p_2 + p_3) / 2$
7. inserir novo $elemento(p_1, p_2, p')$
8. atualizar $elemento(p_1, p', p_3)$
9. inserir novo elemento(po_1, po_2, p')
10. atualizar $elemento_{oposto} = (po_1, p', po_3)$
11. Fim se
12. Senão
13. calcular $p' = ((p_1x + p_2x) / 2, (p_1y + p_2y) / 2)$
14. inserir novo elemento(p_1, p_2, p')
15. atualizar elemento
16. Fim senão
17. Fim se
18. Senão
19. técnica baricentrica
20. Fim senão
21. Fim para

Tab. C.2: Algoritmo de integridade de bordas retangulares.

Na Tabela C.2, a linha 2 refere-se à condição caso o elemento pertença ao conjunto dos elementos que fazem parte da interface. A linha 3 verifica se existe um par oposto ao elemento em questão. Caso exista, devem-se aplicar as linhas de 4 a 10 do algoritmo, onde se criam 4 elementos: p_1p_2p' , $p_1p'p_3$, $po_1p'po_3$,

po_1po_2p' . Caso não exista, visto pela linha 12, apenas 2 elementos serão criados: p_1p_2p' e $p_1p'p_3$. Na linha 18, o elemento é considerado do tipo interno, portanto basta apenas a aplicação da técnica baricêntrica.

C.2.2 Algoritmo de Integridade de Interfaces Circulares

De acordo com a Figura C.4, os ângulos θ_A e θ_B são dados por $tg\theta = \left| \frac{y'}{x'} \right|$.

Sabe-se que a disposição dos elementos, onde uma das arestas faz parte da interface circular, deve pertencer a uma das 8 condições. O par de pontos A e B correspondem a uma dessas possibilidades, onde o primeiro associa-se ao ângulo θ_A e o segundo ao θ_B . Essas condições são:

- Primeiro quadrante:
 - Elemento interno e externo: se $(x_A \geq 0 \text{ e } y_A \geq 0) \text{ e } (x_B \geq 0 \text{ e } y_B \geq 0)$
- Primeiro e segundo quadrante:
 - Elemento interno: se $((x_A > 0 \text{ e } y_A > 0) \text{ e } (x_B < 0 \text{ e } y_B > 0))$
 - Elemento externo: se $((x_A < 0 \text{ e } y_A > 0) \text{ e } (x_B > 0 \text{ e } y_B > 0))$
- Segundo quadrante:
 - Elemento interno e externo: se $(x_A \leq 0 \text{ e } y_A \geq 0) \text{ e } (x_B \leq 0 \text{ e } y_B \geq 0)$
- Segundo e terceiro quadrante:
 - Elemento interno: se $((x_A < 0 \text{ e } y_A > 0) \text{ e } (x_B < 0 \text{ e } y_B < 0))$
 - Elemento externo: se $((x_A < 0 \text{ e } y_A < 0) \text{ e } (x_B < 0 \text{ e } y_B > 0))$
- Terceiro quadrante:
 - Elemento interno e externo: se $(x_A \leq 0 \text{ e } y_A \leq 0) \text{ e } (x_B \leq 0 \text{ e } y_B \leq 0)$
- Terceiro e quarto quadrante:
 - Elemento interno: se $((x_A < 0 \text{ e } y_A < 0) \text{ e } (x_B > 0 \text{ e } y_B < 0))$
 - Elemento externo: se $((x_A > 0 \text{ e } y_A < 0) \text{ e } (x_B < 0 \text{ e } y_B < 0))$
- Quarto quadrante:

- Elemento interno e externo: se $(x_A \geq 0 \text{ e } y_A \leq 0) \text{ e } (x_B \geq 0 \text{ e } y_B \leq 0)$
- Quarto e primeiro quadrante:
 - Elemento interno: se $((x_A > 0 \text{ e } y_A < 0) \text{ e } (x_B > 0 \text{ e } y_B > 0))$
 - Elemento externo: se $((x_A > 0 \text{ e } y_A > 0) \text{ e } (x_B > 0 \text{ e } y_B < 0))$

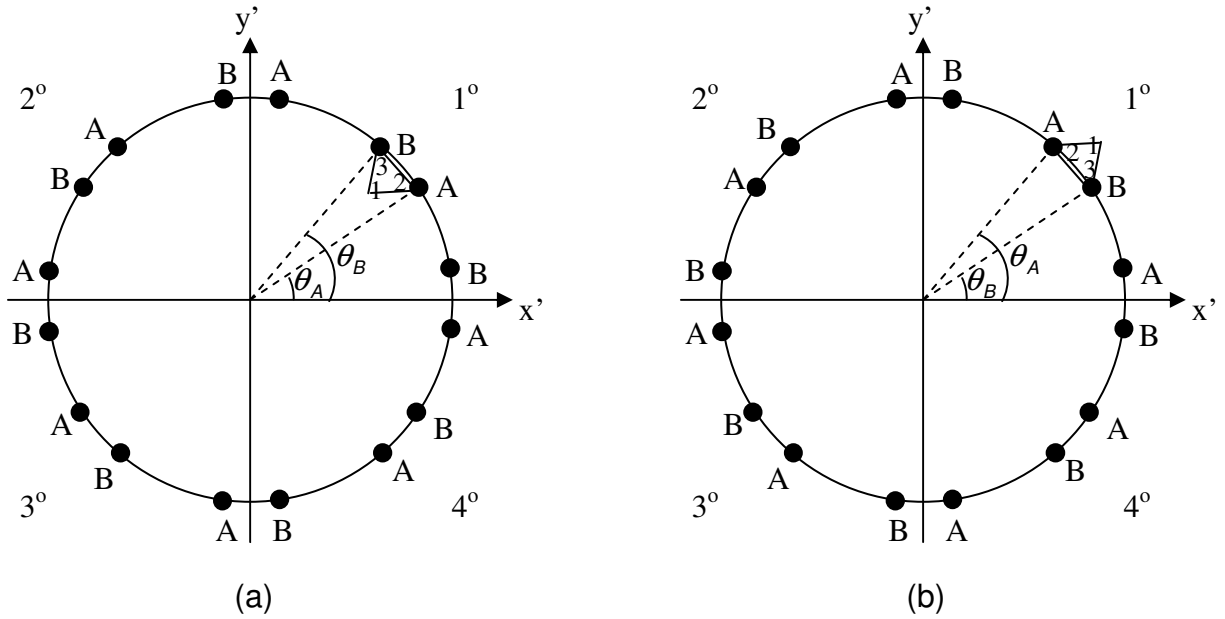


Fig. C.4: Possíveis disposições do elemento na interface. (a) Elemento interno ao círculo, e (b) elemento externo.

Dessa forma, se os vértices do elemento da interface pertencem ao primeiro quadrante, os ângulos serão dados por θ_A e θ_B ; se for primeiro e segundo quadrante, será dado por θ_A e $\pi - \theta_B$; segundo quadrante, $\pi - \theta_A$ e $\pi - \theta_B$; segundo e terceiro quadrante, $\pi - \theta_A$ e $\pi + \theta_B$; terceiro quadrante, $\pi + \theta_A$ e $\pi + \theta_B$; terceiro e quarto, $\pi + \theta_A$ e $2\pi - \theta_B$; quarto quadrante, $2\pi - \theta_A$ e $2\pi - \theta_B$; quarto e primeiro quadrante, $2\pi - \theta_A$ e $2\pi + \theta_B$. A partir disso, o cálculo do ponto *stitch* será dado pelo ângulo médio, $(\theta_A + \theta_B)/2$. O algoritmo de integridade, mostrado na Tabela C.3, segue o mesmo raciocínio das interfaces retangulares. A diferença concentra-se no cálculo do *stitch point*, localizado na linha 15 da Tabela C.2.

Algoritmo: Integridade em interface circular

Entrada: $Elem_{interface} = \{h_1, h_2, \dots, h_n\}$, centro (x_0, y_0) , raio.

1. Para elemento = 1... $N_{elemento}$
2. Se $elemento \in Elem_{interface}$
3. Se elemento possui par oposto que $\in Elem_{interface}$
4. obter pontos do $elemento = p_1, p_2, p_3$
5. obter pontos do $elemento_{oposto} = po_1, po_2, po_3$
6. $x_2 = p_2x - x_0$ e $y_2 = p_2y - y_0$
7. $x_3 = p_3x - x_0$ e $y_3 = p_3y - y_0$
8. $\theta_A = \left| \arctg \frac{y_2}{x_2} \right|$ e $\theta_B = \left| \arctg \frac{y_3}{x_3} \right|$
9. Se $((x_2 \geq 0 \text{ e } y_2 \geq 0) \text{ e } (x_3 \geq 0 \text{ e } y_3 \geq 0))$ //prim.
10. $\theta_A = \theta_A$ e $\theta_B = \theta_B$
11. Senão se $((x_2 > 0 \text{ e } y_2 > 0) \text{ e } (x_3 < 0 \text{ e } y_3 > 0))$
12. $\theta_A = \theta_A$ e $\theta_B = \pi - \theta_B$
13. Senão se $((x_2 < 0 \text{ e } y_2 > 0) \text{ e } (x_3 > 0 \text{ e } y_3 > 0))$
14. $\theta_A = \pi - \theta_A$ e $\theta_B = \theta_B$
15. Senão se $((x_2 \leq 0 \text{ e } y_2 \geq 0) \text{ e } (x_3 \leq 0 \text{ e } y_3 \geq 0))$ //seg.
16. $\theta_A = \pi - \theta_A$ e $\theta_B = \pi - \theta_B$
17. Senão se $((x_2 < 0 \text{ e } y_2 > 0) \text{ e } (x_3 < 0 \text{ e } y_3 < 0))$
18. $\theta_A = \pi - \theta_A$ e $\theta_B = \pi + \theta_B$
19. Senão se $((x_2 < 0 \text{ e } y_2 < 0) \text{ e } (x_3 < 0 \text{ e } y_3 > 0))$
20. $\theta_A = \pi + \theta_A$ e $\theta_B = \pi - \theta_B$
21. Senão se $((x_2 \leq 0 \text{ e } y_2 \leq 0) \text{ e } (x_3 \leq 0 \text{ e } y_3 \leq 0))$ //terc.
22. $\theta_A = \pi + \theta_A$ e $\theta_B = \pi + \theta_B$
23. Senão se $((x_2 < 0 \text{ e } y_2 < 0) \text{ e } (x_3 > 0 \text{ e } y_3 < 0))$
24. $\theta_A = \pi + \theta_A$ e $\theta_B = 2\pi - \theta_B$
25. Senão se $((x_2 > 0 \text{ e } y_2 < 0) \text{ e } (x_3 < 0 \text{ e } y_3 < 0))$
26. $\theta_A = 2\pi - \theta_A$ e $\theta_B = \pi + \theta_B$
27. Senão se $((x_2 \geq 0 \text{ e } y_2 \leq 0) \text{ e } (x_3 \geq 0 \text{ e } y_3 \leq 0))$ //quar.
28. $\theta_A = 2\pi - \theta_A$ e $\theta_B = 2\pi - \theta_B$
29. Senão se $((x_2 > 0 \text{ e } y_2 < 0) \text{ e } (x_3 > 0 \text{ e } y_3 > 0))$
30. $\theta_A = 2\pi - \theta_A$ e $\theta_B = 2\pi + \theta_B$
31. Senão se $((x_2 > 0 \text{ e } y_2 > 0) \text{ e } (x_3 > 0 \text{ e } y_3 < 0))$
32. $\theta_A = 2\pi + \theta_A$ e $\theta_B = 2\pi - \theta_B$
33. Fim se

Tab. C.3: Algoritmo de integridade de bordas circulares.

```

34.       $\theta_{Medio} = (\theta_A + \theta_B) / 2$ 
35.       $x' = raio * \cos(\theta_{Medio})$  e  $y' = raio * \sin(\theta_{Medio})$ 
36.       $x' = x' + x_0$  e  $y' = y' + y_0$ 
37.      inserir novo elemento( $p_1, p_2, p'$ )
38.      atualizar elemento( $p_1, p', p_3$ )
39.      inserir novo elemento( $po_1, po_2, p'$ )
40.      atualizar elementooposto = ( $po_1, p', po_3$ )
41.      Fim se
42.  Fim se
43.Fim para

```

Tab. C.3: Algoritmo de integridade de bordas circulares (continuação).

C.2.3 Algoritmo de Integridade de Curvas

O algoritmo de integridade de curva é mostrado na Tabela C.4. A linha 2 indica, inicialmente, que o elemento deve possuir 2 vértices na interface da curva. Na linha 3 e 4, como estamos considerando que a numeração local 2 e 3 é sempre pertencente à interface, tem-se que as variáveis $param_1$ e $param_2$ correspondem aos valores armazenados do parâmetro t , que indica a localização do vértice sobre a curva. Após o cálculo do novo ponto, mostrado na linha 6, é importante identificar se o elemento corrente, que irá sofrer a quebra, é o elemento à esquerda ou à direita da curva. Isso é realizado na linha 16. Essa identificação é importante para a atualização e a inserção correta de novos elementos gerados. Dessa forma, o mesmo deverá ocorrer para o par oposto ao elemento corrente.

Algoritmo: Integridade em interface de curva arbitrária

Entrada: $Nelem$, $Elem_{interface} = \{h_1, h_2, \dots, h_n\}$

1. Para elemento = $1 \dots N_{elemento}$
2. Se $elemento \in Elem_{interface}$
3. $param_1 = t_{min}$ do elemento
4. $param_2 = t_{max}$ do elemento
5. $noviParam = (Param_1 + Param_2) / 2$
6. calcular $p' = B_n(noviParam)$
7. inserirPonto (B_{n_x}, B_{n_y})
8. obter pontos p_1 , p_2 e p_3 do elemento
9. Se parâmetro t de $p_2 = param_1$
10. $orientXi = p_2x$ e $orientYi = p_2y$
11. $orientXf = p_3x$ e $orientYf = p_3y$
12. Senão
13. $orientXi = p_3x$ e $orientYi = p_3y$
14. $orientXf = p_2x$ e $orientYf = p_2y$
15. Fim se
16. calcular orient $\begin{bmatrix} orientxi & orientyi & 1 \\ orientxf & orientyf & 1 \\ ponto1x & ponto1y & 1 \end{bmatrix}$
17. se orient < 0 //o elemento está direita da curva
18. atualizar elemento(p_1 , p_2 , p')
19. inserir elemento(p_1 , p' , p_3)
20. Senão se orient > 0 //o elemento está esquerda da curva
21. atualizar elemento (p_1 , p' , p_2)
22. inserir elemento(p_1 , p_3 , p')
23. Fim se
24. Fim se
25. Fim para

Tab. C.4: Algoritmo de integridade de interface curva.

C.3 Eficiência do Programa de Elementos Finitos

A fim de obter uma breve compreensão da questão prática do gerador de malhas, esta seção realiza uma descrição de exemplos de artefatos obtidos. Antes de dar início, é importante dizer que conceitos teóricos de modelagem UML e representações de diagrama ontológico não serão discutidos e podem ser encontrados, respectivamente, em [40] e [49].

Com o auxílio da estrutura de classes e da linguagem orientada a objeto, é possível implementar de forma mais simples uma biblioteca geradora de elementos finitos 2-D. Através dessa biblioteca, os algoritmos se tornam mais simples quanto aos aspectos de programação, depuração e manutenção do programa. Como se sabe, as principais desvantagens da utilização da programação estruturada em relação à linguagem orientada a objeto, é que as rotinas complexas não permitem manutenção por módulos, dificulta o entendimento, e prejudica a escalabilidade do projeto. Além disso, os artefatos gerados na orientação a objeto (documentação e diagramas) facilitam a comunicação entre analistas, desenvolvedores e usuários que planejam estudar ou dar continuidade do projeto.

O projeto e modelagem orientados a objeto é uma maneira de pensar sobre problemas utilizando modelos organizados em torno de conceitos do domínio de aplicação. Segue-se, na Figura C.5, o diagrama de classes de um programa de elementos finitos. E, na Figura C.6, apresenta-se um exemplo de diagrama ontológico que tem como pressuposto que todo ambiente social e físico suporta certas formas de ação ou comportamento.

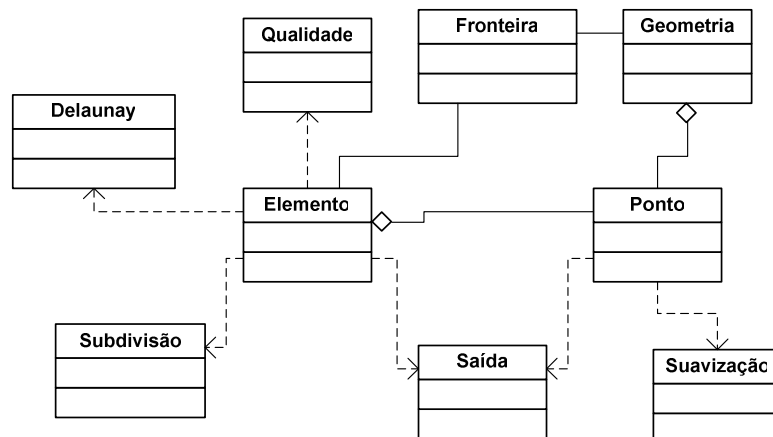


Fig. C.5: Exemplo de diagrama de classes.

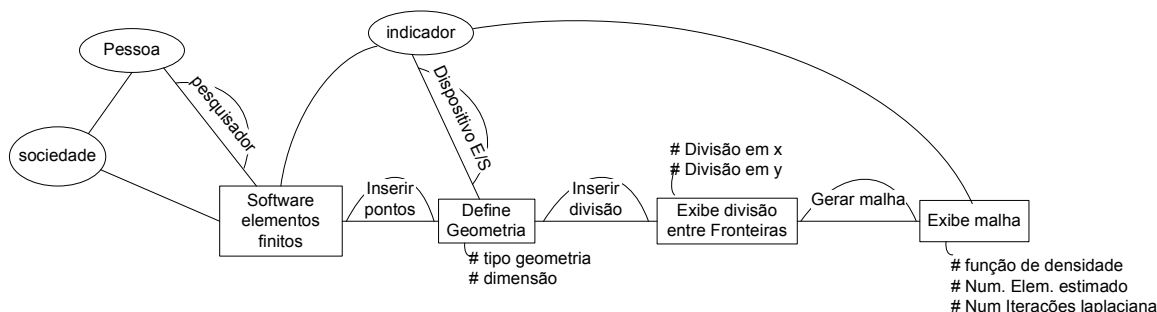


Fig. C.6: Exemplo simples de diagrama ontológico.

Neste estudo também se realizou uma comparação entre o uso de uma lista e de um vetor de objetos, que pode ser elementos ou pontos. Uma lista de elementos consome tempo de processamento devido à referência do ponteiro para outro elemento e, assim, sucessivamente. Supõe-se a geração de 5000 elementos. Para se percorrer desde o primeiro elemento ao último gerado, o tempo consumido é em minutos, enquanto que o uso de vetor já possui uma referência direta para a posição do vetor e seu conteúdo, e seu tempo de processamento é de milissegundos. A vantagem da lista é a possibilidade de alocar dinamicamente a quantidade de elementos, enquanto no vetor é necessário passar essa informação prévia. Dessa forma, o uso de vetor pode conduzir à falta de memória durante a execução do programa. No entanto, como o objetivo é a geração em torno de 50000 elementos, no máximo, o uso de vetores oferece a vantagem quanto ao tempo de espera.

Outro modo que possibilita complementar a eficácia do programa é a utilização da linguagem C++ integrada ao OpenGL, como tem sido utilizada em alguns programas como o GiD [35]. Realizaram-se testes em ambiente C++ integrado ao OpenGL e na linguagem Java. A maioria dos testes dos algoritmos implementados nesse estudo foram realizados no ambiente Java, sem a utilização de rotinas complexas da biblioteca Java. O desempenho é semelhante quanto à solução 2D do problema. No entanto, a facilidade para se trabalhar com um ambiente 3D e modelagem 3D de objetos, utilizando OpenGL, torna imprescindível o uso de C++ e OpenGL ou Java e OpenGL. De acordo com os testes realizados, observou-se que o primeiro é uma boa opção e, também, se mostrou bastante estável no trabalho de Hsu [49]. Quanto à segunda opção, observa-se que as bibliotecas OpenGL para Java encontram-se mais limitadas na internet e além disso, algumas características de programação dificulta o seu uso. Em Java a interface gráfica 2D, para a realização de teste, mostrou-se bastante estável. Embora o foco desse estudo não tenha sido sobre o ambiente gráfico, que o torna menos amigável, foi possível realizar alguns tipos de auxílio, como zooms, deslocamento da imagem, entre outros.

Apêndice D

Saída de Arquivos

Nesta seção, mostra-se um exemplo de saída de arquivo do gerador de malhas, formatado no modelo para simulação numérica no Aníbal.

D.1 Simulação no Aníbal

A saída de dados deve ser compatível com a entrada de dados do programa Aníbal, desenvolvido pelo prof. Dr. Hugo H. E. Figueroa, para simulações modais de guias ópticos. Um exemplo de saída pode ser visto na Tabela D.1. A saída, de forma ordenada, deve ser composta por: número total de nós, coordenadas x e y dos nós ordenados; número total de elementos seguido, da ordem linear (valor 3, e 6 para valor quadrático), composição de cada elemento por meio de 3 vértices (para valor 3): ponto1, ponto2, ponto3, cada um seguido do seu respectivo material; o número total de nós de contorno da PML, identificação desses nós de contorno com identificação de localização na vertical ou horizontal, respectivamente, V e H; número total de materiais na estrutura e seus valores de representação ordenados; número de marcas PML, e a identificação do nó de início da margem superior esquerda e inferior direita, respectivamente, SE e ID.

```

*coordenadas dos nós
1173
1      -1.0000000e+00    1.00000000e+00
2      -1.0000000e+00    9.3750000e-01
3      -9.3750000e-01    1.00000000e+00
...
1171  1.0000000e+00    -9.3750000e-01
1172  9.3750000e-01    -1.0000000e+00
1173  1.0000000e+00    -1.0000000e+00
*matriz de conectividade e vetor de registro
554  3
1      899    799    905    1
2      1173  1169  1170    2
3       1     6     5     2
...
552  489    412    508    1
553  799    759    905    1
554  12     5     6     1
*nós de contorno
*número de nós de contorno
128
*nós de contorno condição
1      1     1     PE    V
2      2     1     PE    V
3      3     1     PE    H
...
126  1171  1     PE    V
127  1172  1     PE    H
128  1173  1     PE    H
*lista das características
2
1      1.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0 1.0 0.0
2      1.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0 1.0 0.0
*nós marcas de PML
2
*nós de contorno condição
1 SE
1173 ID

```

Tab. D.1: Exemplo de saída de arquivo do gerador de malha para simulação.

D.2 Visualização da Malha

Além da malha gerada no programa, a saída do arquivo gerado também pode ser integrada com o programa GiD, considerando apenas a questão da visualização da malha, simplesmente pelo GiD possuir alguns recursos visuais avançados. O arquivo deve ser importado no formato GiD *mesh*. O formato de saída da malha, observado na Tabela D.2, se dá através da seguinte forma:

```
escape escape escape escape
x1 y1
x2 y2
x3 y3
...
xn yn
escape escape
escape escape escape escape
escape escape escape escape geometry create Line

*elemento 1
join
ponto1
ponto2
escape
ponto2
ponto3
escape
ponto3
ponto1
escape
escape escape escape escape
...
* elemento nelem
join
ponto1
ponto2
escape
ponto2
ponto3
escape
ponto3
ponto1
escape
escape escape escape escape
```

Tab. D.2: Exemplo de saída de arquivo para visualização.

Na Tabela D.2, x_n e y_n são as coordenadas do nó n num plano; e ponto1, ponto2 e ponto3 significam os três vértices que compõem um elemento *nelem*.

A Figura D.1 mostra um resultado de visualização de um guia óptico retangular no GiD.

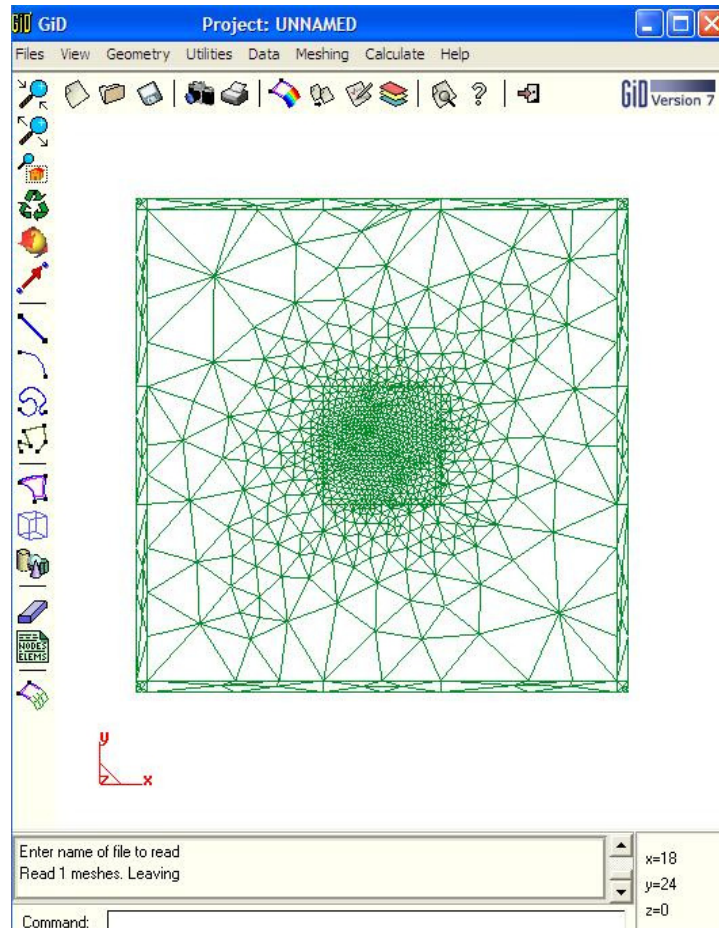


Fig. D.1: Visualização no GiD.